## EXHIBIT A

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | CROSS-REFERENCE TO RELATED APPLICATIONS | P1/L3 |
| **0001** | This application claims benefit of U.S. provisional application 60/273,862, filed March 5, 2001 and having the same title and inventor as the present application. That provisional application, including all materials incorporated by reference therein and its appendices, is incorporated herein by reference in its entirety. Also incorporated herein by reference are the entire contents of this application's specification as originally filed. | P1/L4-7 |
| | BACKGROUND OF THE INVENTION | - |
| **0002** | With the "Electronic Signatures in Global and National Commerce Act" of 2000, the U.S. Congress gave digital signatures the same legal validity as an ink signature on a piece of paper. Now, the sender of an email message, word processing document, or any other type of electronic record that can be construed as a written contract can be legally bound to that record if the recipient can prove that the sender authenticated the record. | C-1, para. 1 |
| **0003** | Electronic records that are signed with digital signatures can be proved, to a very high level of certainty, to be authenticated by the person who caused the digital signature to be applied to the record. The digital signature can only be applied with a private key, which is an incredibly large number that uniquely corresponds to another incredibly large number, called a public key. The private key, as its name implies, is kept a strict secret by the person who uses it to sign his or her digital signature. Strong cryptographic software ensures that it is "computationally infeasible" (i.e., very difficult, even with very fast computers) to derive the private key from the public key. When a person signs an electronic record with their private key, a digital signature code is produced that anyone can verify against the public key, which is publicly accessible. The slightest change in a document so signed will cause the digital signature to no longer match the document. | C-1, para. 2 |
| **0004** | The cryptography used in digital signatures is very strong and nearly impossible to tamper with, at least with current technology. But a very old problem remains that technology alone cannot entirely solve. That problem is trust. | C-1, para. 3 |
| **0005** | The trust problem in digital signatures can be summarized as follows: How do you know that the public key really belongs to the person who says it belongs to him or her? Anyone can create a public | C-1, para. 4;<br>C-2, para. 1<br>*(bracketed* |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | key and call it someone else's, then use the corresponding private key to create forged electronic records. The 1998 edition of The Global Trust Register, a printed directory of public keys published by a group of cryptography experts, states the problem as follows: "[T]here is no cheap and effective way for Internet users to check the validity of public keys on which they may wish to rely." | *"[T]here" is as it appears in Appendix C)* |
| **0006** | The experts who wrote The Global Trust Register made that statement in spite of the many efforts by Certification Authorities (CAs) to deploy a "hierarchical trust" model, where trusted third parties check out the identity of persons who own private/public key pairs. A CA such as VeriSign, Entrust, or Thawte will add its digital signature to a public key if the public key is tied to the name of a person who physically appears with proper documentation to prove their identity. Recipients of documents signed with the certified public key are then expected to trust that the CA has done its job and that the public key really came from the person whose name is tied to it. | C-2, para. 2 |
| **0007** | But what happens when one of the many employees at the CA doesn't do his or her job properly? Who is liable for the recipient's reliance on a forged document promising delivery of 10,000 widgets for $1,000,000 when the sender has pocketed the money and run, completely anonymously due to the faceless nature of the Internet? The recipient cannot sue the sender if the recipient doesn't know the sender really was. The recipient's only course of action is to sue the CA for not doing its job. CAs try to avoid liability with disclaimer language in their Certification Practice Statements. | C-2, para. 3 |
| **0008** | What about tort claims against the CA? Here's what the text Certification Authority Liability Analysis has to say about that: "_A CA's liability for tort claims based on negligence may be limited by the so-called ["]'economic loss doctrine.["]' The economic loss doctrine provides that claims for purely economic losses based on product defects are not recoverable in tort. The rule holds simply that tort liability does not arise for pure economic loss, but only for personal injury or property damage. The principles behind this rule are that protecting personal injury and property damage claims are more important social policies than pure economic (business) losses, and that economic losses are better protected by negotiated contract allocations rather than through generalized tort law" (Certification Authority Liability Analysis Section 1.1, American Banker's Association, 1998). | C-2, para. 4, including block quote |
| **0009** | In addition to the problems with "hierarchical trust" that should now be apparent, reliance on the Certification Authority as a trusted third | C-2, first para. after block |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
|  | party requires the CA to have an established reputation and to keep its digital house in order for a long time. It doesn't do much good to have a "trusted" third party certifying a digital signature if that third party disappears, loses data, or is found out to have some serious security breach in its infrastructure. | quote |
| **0010** | In view of these problems, a system is needed that will translate the direct trust from signer to recipient that self-authenticating ink signatures now provide into the realm of digital signatures. The solution, ~~it turns out~~ applicant has discovered, is combining technology with the trusted authentication that ink signatures and signature witnesses have established over hundreds of years of history. | C-2, last para.;<br>C-3, para. 1 |
| **0011** | ~~According, there is a~~ Another need addressed by the inventions is for a system of destroying electronic communications or records when the sender and recipient of the communications agree to do so. | AI-1, para. 2 |
| ... | In private confidential conversation, two people can have a conversation without leaving any record of their conversation. With written or electronic communications, however, there is some record of what was said. That record can be difficult to eliminate. While paper communications such as letters can be shredded if both sender and recipient agree that they will destroy their copies, electronic communications (e-mail) are more difficult to eliminate because backup copies can be made and automatically archived onto other locations. It is sometimes surprising that backup copies are available during discovery of communications that would be embarrassing. | AI-1, paras. 1-2 |
| **0012** | ~~A pseudogroup operation according to various aspects of~~ Another need addressed by the inventions is dispens[es]ing with the need for the modulus | P14/L10-11 |
| ... | in the multiplicative group xy modulo p | P13/L25-26 |
| ... | to be fixed with respect with respect to the order of the input and output set. | P14/L11-12 |
| ... | The IDEA cipher ~~exploits this property by using~~ uses a multiplicative group modulo p = $2^{16} + 1$ (which is prime) along with two other group operations to encrypt binary data in the set ~~{1, 2, ...~~ [$2^{16}$}] of 16-bit integers, but | P14/L4-5 |
| ... | very few known moduli have the desirable property of being exactly one greater than a power of two. ~~Appendix J 16 illustrates the~~ The result is an undesirable lack of scalability ~~in the IDEA cipher that results from this fact~~. | P14/L7-9 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| **0013** | Other needs addressed by the inventions of this application include providing a simple, intuitive way of authenticating an electronic record, | P12/L5-6; AD-1, para. 1 |
| ... | making digital signatures ~~can be made~~ unobtrusive, | P14/L10-12 |
| ... | and increas[ed]ing intractability to an attacker without creating any noticeable inconvenience for [the] a legitimate user. | P10/L17-18 |
|  | SUMMARY OF THE INVENTION | - |
| **0014** | A ~~the~~ “covenant trust” authentication system of the inventions employs a printed [1.] authorization and certification instrument (ACI) that, with an ink signature, legally binds the signer to digital signatures uniquely corresponding to (and thus created with) a positively identified public key. | P2/L25-27;<br>P3/L1 |
| ... | The key code in the ACI can be printed throughout the background of the entire paper as vertically oriented digits in various outline fonts. | C-9, bullet para. 1 |
| **0015** | A method of the invention for “self-certification of digital signature keys by contract” uses a | C-1, line 3 |
| ... | ~~the~~ covenant trust model of key certification. | C-1, line 5 |
| ... | The method certif[ying]ies a subscriber's public key by ~~comprising~~ (a) having the subscriber execute an ACI; (b) confirming that the ACI has been signed and notarized in ink; and (c) publishing the confirmation for persons relying on the public key. | P3/L3-5 |
| **0016** | According to various aspects of the present inventions, an electronic record (e.g., MICROSOFT WORD document, AUTOCAD drawing, etc.) can be signed by “printing” that record to a special “virtual signature printer,” ~~An example of a virtual printer is the "PDF Writer" printer driver that is installed with the ADOBE ACROBAT software. The inventive "virtual signature printer"~~ which provides the user with an intuitive, simple way of authenticating an electronic record. The “virtual signature printer” system produces an output file (or multiple files~~, see below~~) that can be sent to a recipient for viewing, printing, and validation. The recipient can view or print the file ~~(preferably, the file is “backward compatible” compatible with widely available viewing software)~~ and, with special software, can validate the signature on the file. | AD-1, paras. 1-2 |
| **0017** | A particularly advantageous way of signing an electronic record that has been “printed” this way is with embedded signatures. | AD-1, para. 3 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| ... | Per one embedded signature[:] aspect, graphical indicia of a signature are placed within a graphical (image-based) representation of an electronic record. | P12/L7-8;<br>P2/L2 *(aspects vs. numbered paras.)* |
| ... | Per another embedded signature[:] aspect, signature data is placed within an unused or backward-compatible field of an electronic record. | P12/L9-10;<br>P2/L2 *(aspects vs. numbered paras.)* |
| **0018** | ~~The following terms are said to be~~ An element[s] of the inventions | P18/L5 |
| ... | ~~"Secure Delay."~~ is a processing-induced secure delay interposed between an input and output that transforms a given input value into an unpredictable but deterministic output value. | P22/L12-14 |
| ... | A delay system according to another aspect of the invention~~, illustrated in the block diagrams of Z7 and Z8,~~ makes a secure delay according to various aspects of the invention less obtrusive to the user~~. It does so~~ by beginning the delay process when the user's passphrase has been partially entered. Advantageously, such a system performs the delay computations substantially in parallel with the unavoidable delay of the user's input of the passphrase. | Z-1, para. 3 |
| **0019** | A secure delay according to various aspects of the present inventions can be applied in other areas than just passphrase security. For example, a hash value can be run through a secure delay to produce a smaller hash value that would be computationally infeasible to derive based on a birthday attack. | AA-1, para. 1 |
| **0020** | ~~(As but one example, the applicant views~~ The implementation of a secure delay with a pseudogroup operation~~ as~~ is a particularly advantageous combination according to one aspect of ~~his~~ the inventions. ~~See Appendices K and L for specific disclosure of this particular combination.)~~ | P2/L15-18 |
| ... | A pseudogroup operation according to various aspects of the present inventions dispenses with the need for the modulus p to be fixed with respect to the order of the input and output set | P14/L10-12 |
| ... | in an operation of the type xy modulo p. | P13/L25 |
| ... | Advantageously, the modulus can be chosen as whatever prime number is closest to the set order - above it or below it. | P14/L12-13 |
| **0021** | A cryptographic document destruction | AI-1, title |
| ... | [A] system according to various aspects of the invention includes[:] an encryption subsystem[;] and a decryption subsystem[,]. The | AI-1, para. 3 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | decryption subsystem ~~using~~ uses a temporary key that can be disposed of to make encrypted communications unreadable. | |
| ... | Per one aspect, periodic purging of documents ~~and/~~or e-mail messages, or both, is done by mutually agreed ~~and/~~or scheduled destruction of shared keys, or both. | P13/L13-14 |
| ... | Per another aspect, a hardware electronic record "lock" with a disposable key token is employed, such as a simple paper card with a bar code printed on it. ~~See Appendix AJ.~~ The key token can be thrown away when the "locked" (i.e., encrypted) electronic record is to be purged, ~~by~~ making decryption of it practically impossible. | P13/L15-18 |
| **0022** | ~~Indeed,~~ The applicant contemplates that his inventions~~, as supported by the disclosure of this informal application and appendixed drawings,~~ include all systems and methods that can be practiced from all suitable combinations of the various aspects disclosed[,] and ~~all suitable combinations of the~~ exemplary elements listed. | P2/L11-15 |
| ... | Such combinations have particular advantages, including advantages not specifically recited herein. | P2/L18-19 |
| | BRIEF DESCRIPTION OF THE DRAWINGS | P1/L8 |
| **0023** | ~~See~~ FIG. 1 is a flow diagram of a signer enrollment procedure of an exemplary self-certification process. | P3/L6; B-1 *("SIGNER ENROLLMENT")* |
| **0024** | FIG. 2 illustrates a signing and verif[y]ication procedure of the exemplary self-certification process of FIG. 1. | B-2 *("SIGN + VERIFY")* |
| **0025** | FIG. 3 illustrates an exemplary process ~~is illustrated in the attached FIG. (AF-4)~~ | AF-1, para. 4 |
| ... | for signing a document ~~can be signed~~ by printing ~~the document~~ to a TIFF file using a virtual printer driver, | AF-1, lines 2-3 |
| ... | with a signature block contained within a suitable field of the TIFF file. | AF-1, lines 7-8 |
| **0026** | FIG. 4 illustrates an exemplary process ~~is illustrated in the attached FIG. (AE-4), in which a signer creates~~ for digitally signing a document | AE-1, para. 3, lines 1-2 |
| ... | ~~He then prints the document to the SelfCertify.com~~ by virtual print[er]ing | AE-1, para. 3, lines 3-4 |
| ... | to a TIFF file with a graphical signature block. | AE-4 *(FIG. 4, ref. 418)* |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| **0027** | ~~This brief disclosure is~~ FIG. 5 illustrates part of an example ~~of a clear-signed~~ document to be signed | AE-1, para. 1 |
| ... | ~~The user can specify the region by moving~~ with a dashed box appearing over text of the | AE-1, last para.;<br>AE-3 |
| ... | ~~1. Selection of a graphical region within the displayed~~ document for application of a signer's digital signature "stamp[.],"at a ~~The~~ user-specified ~~can specify the~~ region. ~~by moving around the screen. . . . An unsigned signature page of this letter is attached showing what such a box might look like as it is moved around during the selection process~~. | AE-1, last para.;<br>AE-2, first para. |
| **0028** | ~~This brief disclosure is an example of a clear-signed~~ FIG. 6 illustrates the document of FIG. 5 with a simulated digital signature ~~(25x13 bits = 325 bits)~~ that resides a graphic within a signature block that is excluded from the signature calculation of the document. | AE-1, para. 1 |
| **0029** | ~~This is~~ FIG. 7 illustrates an example of a message that has been signed with preserved formatting and an unobtrusive digital signature around clear-signed text. | AH-1 |
| **0030** | FIGS. 8-12 ~~Z2-Z6~~ illustrate screen shots of ~~a secure passphrase entry system according to various aspects of the invention, illustrating~~ an exemplary user interface at different points during the input of a passphrase without the use of keystrokes. | Z-1, lines 2-4 |
| **0031** | FIGS. 13-14 illustrate ~~24.~~ graphical (mouse) input of "Alphanumeric-Except-O" passphrase digits in a 7x5 grid, with already-selected selected digits indicated. ~~See Appendix AB.~~ | P12/L14-15 |
| **0032** | FIGS. 15-16 are block diagrams illustrating a delay system according to an~~other~~ aspect of the invention~~, illustrated in the block diagrams of Z7 and Z8~~. | Z-1, para. 1 |
| **0033** | FIG. 17 illustrates one exemplary embodiment of a 32-bit secure delay employing a [α is] generator α of $Z_p$* ~~(fixed)~~, where p is prime and between the values $2^{32} - 2^k$ [< p <] and $2^{32}$. | AC-2 *(Note at lower right of page; FIG. 17, ref. 1720 and note above it)* |
| **0034** | FIG. 18 illustrates a ~~modified "⊖"~~ pseudogroup operation from inputs x & y to an output z with $p > 2^N$. | J-1 *(Note at upper right of page; label "With $p > 2^N$"; input, output labels)* |
| **0035** | FIG. 19 illustrates an inverse pseudogroup operation from inputs z & $y^{-1}$ to an output. | J-1 *(Label "INVERSE:"; output labels)* |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| **0036** | FIG. 20 illustrates a pseudogroup operation from inputs x & y to an output z with $p < 2^N$. | J-1 *(Label "With $p < 2^N$"; input, output labels)* |
| **0037** | ~~27. Periodic~~ FIG. 21 illustrates purging of ~~documents and/or e-mail~~ a sensitive message by a mutual~~ly agreed and/or scheduled~~ destruction of shared keys. | P13/L13-14;<br>AI-1 *(FIG. 21, refs. 2112, 2144; refs. 2122, 2152 → refs. 2124, 2154)* |
|  | DETAILED DESCRIPTION | - |
| **0038** | The "covenant trust" authentication system employs an [1.] authorization and certification instrument ("ACI") that, with a holographic (ink) signature, legally binds the signer to digital signatures uniquely corresponding to (and thus created with) a positively identified public key. | P2/L25-27;<br>P3/L1 |
| **0039** | ~~See~~ An example is shown in Appendix A[-1] of the application as originally filed. It includes the following text superimposed on vertically-oriented background digits "FC1D 9CBF B80D E940 6250 7CAD 070F 90BA CDA2 44DO" in various fonts and sizes: | P2/ P-19, A-1 *(see background digits)* |
| **0040** | "Using a PGP-compatible cryptosystem, I have created a 'DH/DSS' private key ('the Private Key') and public key ('the Public Key'). The Public Key uniquely corresponds to a 160-bit 'fingerprint' (SHA-1 hash). The fingerprint is printed below as ten groups of four hexadecimal digits, and in various outline fonts throughout the background of this entire paper and in all its signature fields: FC1D 9CBF B80D E940 6250 7CAD 070F 90BA CDA2 44D0. | A-1, lines 3-6 |
| **0041** | "The Private Key uniquely corresponds to the Public Key but is computationally infeasible to derive from the Public Key. (Two data sets are said to 'uniquely correspond' when it is computationally infeasible that data sets other than those uniquely corresponding could be found to match, using the cryptographic process that is intended to match the data sets.) From time to time, I expect to apply my digital signature to electronic records, with the same legal effect as my ink signature would have on a printed copy thereof. When I do so, a set of signature data is produced that uniquely corresponds to both the document being signed and the Public Key. The slightest change in a document so signed will cause the digital signature to no longer match it. | A-1, lines 7-12 |
| **0042** | "By executing this authentication and certification instrument ('ACI'), I acknowledge, stipulate, and covenant that any electronic record accompanied by a digital signature that uniquely corresponds | A-1, lines 13-19 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
|  | to both the document and the Public Key (in accordance with the Digital Signature Standard) was signed by me, beyond any reasonable level of doubt. I COVENANT WITH ANY BEARER OF THIS AUTHORIZATION OR FACSIMILE COPY THEREOF NOT TO REPUDIATE SUCH DIGITAL SIGNATURE UNLESS I COMMUNICATE A REVOCATION OF THE PUBLIC KEY TO THE BEARER IN WRITING BEFORE THE SIGNATURE DATE. (A 'writing' is an electronic record signed with the digital signature or a paper signed with my ink signature.) A revocation will be deemed constructively communicated upon publication at SelfCertify.com. |  |
| **0043** | "I agree to always keep the Private Key encrypted (except during brief intervals of use) with a secure passphrase that only I know, to never reveal the passphrase to anyone (except as compelled to do so by law), and to use the Private Key only on trustworthy computer systems. I understand that the consequence of failing to do so is the possible forging of my digital signature, which is subject to the non-repudiation covenant above. I understand that to be trustworthy, a computer system must employ computer hardware, software, and procedures that: (1) are reasonably secure from intrusion and misuse; (2) provide a reasonably reliable level of availability, reliability, and correct operation; (3) are reasonably suited to performing their intended functions; and (4) adhere to generally accepted security principles. (Digital Signature Guidelines, American Bar Association, 1996.) I have read, understand, and agree to follow guidelines provided to me by SelfCertify.com for selecting a secure passphrase and for ensuring that all computer systems on which I use the passphrase remain trustworthy. | A-1, lines 20-28 |
| **0044** | "Any ACI (or facsimile copy thereof) including (1) language substantially identical to that above, (2) the fingerprint (or key value) of any other public key, and (3) what is purported to be my notarized handwritten signature is presumed to be a forgery unless it is accompanied by a written revocation of the Public Key of this ACI or a digitally signed ACI of the other key, signed with the Public Key of this ACI. The written revocation can be signed with my original, notarized ink signature or digitally with the Public Key of this ACI." | A1, lines 29-33 |
| **0045** | The background digits are of a ["]fingerprint[."], a number typically fairly long (e.g., 160 bits), uniquely corresponding to an electronic record (e.g., a signing key, encryption key, dual function key, computer file, etc.) Often, a fingerprint is simply a hash (e.g., SHA-1) of the electronic record. According to aspects of the present inventions, a fingerprint can be encoded into a user-memorable form, such as that used in the pronounceable passphrase aspects of the inventions, or a familiar-type number resembling a telephone | A-1 *(background digits)*;<br>P19/L8-14 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
|  | number, ZIP code, etc. ~~See Appendix AA and Item 12 above.~~ |  |
| **0046** | The example ACI further includes signature, notarization, and certified copy portions, also superimposed on the background digits. The notarization portion includes the following statement: "This (blank) day of (blank), (blank), the above-named person personally came before me, produced a mailing label of an overnight delivery package listing him or her as the addressee and two forms of identification (including one with a matching photograph). The person then executed the foregoing ACI in my presence. On the line below, I have written the mailing or tracking number of the mailing label. I will mail this signed document directly to SelfCertify.com at the return address shown on the mailing label." | A-1 *(bottom part)* |
| **0047** | The ["]ACI~~," "Authorization and Certification Instrument"~~ is a legal instrument executed by a conventionally accepted technique (e.g., handwritten ink, a witnessed audio or video recording, etc.) that does not rely on digital signature technology. The instrument includes a covenant not to repudiate a particularly identified digital signature key of the signer, making electronic records authenticated with that digital signature key as valid (assuming the court accepts the certification) as the conventional signature on the ACI. ~~See Appendix A and C 3.~~ | P18/L11-17 |
| **0048** | ~~Digits that~~ The ["]background digits[."] of the example ACI ~~that~~ can be read in the background of a document "behind" other alphanumeric indicia. In a particularly advantageous arrangement of the example ACI, background digits are oriented perpendicular to the main orientation of the document. ~~(See the large Background Digits in Appendix A)~~ In another particularly advantageous arrangement of the example ACI, small background digits fill a substantial portion of one or more document signature, stamp, and/or annotation fields. ~~(See Appendix A)~~ In the example, the small background digits in the signature fields where "void"is handwritten in pace of actual signatures. | P18/L18-25 |
| **0049** | Background digits according to various aspects of the inventions advantageously maintain a contextual thread between (1) inked indicia (e.g., one or more holographic signatures, a notary stamp, etc.) in a document and (2) printed indicia elsewhere in the document, making it difficult to "lift" the image of such inked indicia and transfer it to a forged document. The background digits are preferably in an outline font to maintain readability of the main document indicia. The spacing, font size, and font type of the digits is preferably varied in a way unpredictable to a forger (e.g., pseudorandomly) to make duplication of a "erasing negative" of the | P19/L25-28;<br>P20/L1-7 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | background digits difficult. (An erasing negative could conceivably be used to eliminate background digits and permit "lifting" of inked indicia that would otherwise have digits in its background field.) | |
| **0050** | As discussed above, an ACI binds a signer to a digital signature ["]uniquely corresponding[."] ~~See Appendix A.~~ to a positively identified public key. Most fingerprints and digital signatures could conceivably correspond to multiple records. However, the likelihood of finding a corresponding electronic record other than the one of interest, given a uniform probability of obtaining all possible fingerprints of digital signatures from a given record, is usually vanishingly small. The likelihood of finding a second match that could be mistaken for the electronic record of interest is so small, in most cryptographic applications, as to be considered impossible. Embodiments are certainly possible, however, in which and electronic record considered to uniquely correspond to a fingerprint or digital signature with a higher probability of a false match. | P23/L1-10 |
| **0051** | The ACI ~~(Appendix A)~~ is a specific example of a positive identification of an electronic record in which the electronic record is a public signing key. (Another example in which an electronic copy of a publicly accessible paper file is positively identified by a third party who has inspected the file, is shown in Appendix F.) | P4/L18-21 |
| ... | ~~5. Positive~~ The identification ~~of an electronic record with~~ employs an "integrated" combination of: (a) a code "uniquely corresponding" to that electronic record (e.g., a SHA-1 cryptographic hash code); and (b) a holographic signature and (or facsimile thereof). The combination is said to be integrated when it would be difficult for a forger to separate the elements of the combination. Another way of describing and integrated combination of (a) and (b) is having a contextual thread between (a) and (b). (See the discussion of ~~the exemplary element~~ "background digits" ~~below~~ above.) | P4/L3-9 |
| **0052** | A paper document of facsimile copy thereof containing the combination can include the following advantageous aspects: (1) the digits of the code can be printed in background digits of the document, including behind fields for handwriting~~. See Appendix A for an~~, as in the above example ~~of an~~ ACI with background digits of a PGP "fingerprint[.]"; (2) the document can include a facsimile copy of a photographic identification of the signer, which can be referenced in language of the document (e.g., a Notary's statement). For example, an ACI can include a photocopy of a driver's license. | P4/L9-17 |
| **0053** | A method for self-certification of digital signature keys by contract | C-1, line 3 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| ... | uses a covenant trust model of key certification. | C-1, line 5 |
| ... | The method ~~of certifying~~ certifies a subscriber's public key ~~comprising~~ by (a) having the subscriber execute an ACI; (b) confirming that the ACI has been signed and notarized in ink; and (c) publishing the confirmation for persons relying on the public key. ~~See flow diagram of an exemplary process in Appendix B and~~ A description of the method, ~~in Appendix C, Appendix C is adapted from a confidential document~~ describing benefits of the inventive method (as a technology rather than as a product or service) generally and in conjecture only, and regarding a specific example of the method, follows. The specificity of the example is not intended to any way limit the scope of the invention or imply that any product or service described ~~in Appendix C~~ has been offered for sale as of the application's filling date. | P3/L3-12 |
| **0054** | In the method, the covenant, [-] an ancient concept, is applied to technology. The covenant trust model relies on a person's self-certification of his or her public key and a covenant by that person not to repudiate the public key. The "Covenant of Non-repudiation" legally binds the owner of the public key to any digital signatures created with the corresponding private key. Thus, the liability for proper usage of the private key is placed on the shoulders of the person owning the public key, where it belongs, and legal reliance can be placed upon the public key and any electronic record signed with the corresponding private key. | C-3, Subheading II plus following para. 2 |
| **0055** | The covenant is made in an Authentication and Certification Instrument (ACI), a legally signed paper document that contains an identification code positively identifying the public key in question. The document is signed in ink and witnessed by a notary public, thus invoking an authentication system whose trust has been established and is universally recognized by our legal system. | C-3, para. 3 |
| **0056** | ~~An~~ The example ACI ~~(see Appendix A-1)~~ discussed above contains the following text: "I acknowledge and understand that the consequence of executing this authorization and certification instrument ("Authorization") is that any electronic record accompanied by a digital signature that uniquely corresponds to both the document and the Public Key was signed by me, with a negligible level of doubt. I covenant with any bearer of this Authorization or facsimile copy thereof not to repudiate such digital signature unless I communicate (directly or indirectly) a revocation of the Public Key to the bearer in writing before the signature date." | C-3, para. 3 plus following block quote |
| **0057** | The ACI includes security features~~, discussed below,~~ that make it | C-3, first para. |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | extremely difficult to forge with identification of a different public key, even in a facsimile copy. A person receiving a copy of the ACI (from the signer, from the Internet, wherever) is in possession of a legal instrument that authenticates a public key without the need for trusted third parties. The role of a third party, if one is used at all, is simply to distribute facsimile copies of the ACI. For additional security, the third party can apply its digital signature to the copies of the ACI it distributes to certify them as true copies of the original signed in ink. For example, the third party can authenticate PDF or TIFF files containing facsimile copies of ACIs with a standard SSL (Secure Sockets Layer) certificate issued by a conventional CA. | after block quote |
| **0058** | The conventional "hierarchical trust" model attempts to establish a chain of authenticity to supposedly trusted third parties who are presumed to be doing their jobs properly. In contrast, the covenant trust model establishes a chain of authenticity to a legal covenant, signed with a notarized ink signature on an ACI, in which a public key owner promises not to repudiate digital signatures corresponding to that public key. The chain of authenticity can begin with initial reliance on the security features of a facsimile copy of the ACI and distribution of the ACI via a trusted web site, email sender, or remote-access viewing software. Higher up on the chain of authenticity, and still convenient to obtain, is digitally-signed certification of the copy by a trusted certifier. Still higher on the authenticity chain is the availability of ink-signed certified copies of the ACI by the original signer or, for a fee, by a trusted certifier. The ultimate link in the chain of authenticity can be provided by making the original notarized, ink-signed ACI paper available for inspection by experts, judges, juries, or attorneys during dispute resolution. | C-3, last para.;<br>C-4, para. 1 |
| **0059** | The following describes implementation of covenant trust via the Internet. | C-4, Heading B |
| **0060** | Overview | C-4, Subheading II |
| **0061** | A new type of "Certification Authority" ~~will~~ is discussed herein as being deployed at SelfCertify.com based on the covenant trust model. | C-4, para. 2 |
| ... | ~~Appendix C~~ (This portion of the application ~~is adapted from a confidential document describing~~ describes benefits of the inventive method ~~(as a technology rather than a product or service)~~ generally and, in conjecture only, a specific example of the method. The specificity of the example is not intended to in any way limit the scope of the invention.) | P3/L6-10 |

| Para. | Substitute Specification Text with Changes from Source Material | Source Material |
|---|---|---|
| ... | SelfCertify.com (discussed here in the present tense for convenience) is a certification authority only in the sense that it registers public keys and the identity of persons who claim to own those keys, and certifies that copies of ACIs it distributes are true copies of originals in its possession. It does not certify the identities of the person claiming to own the public keys - those persons make that certification themselves in the ACI. | C-4, para. 2 |
| **0062** | In addition to registering public keys and distributing ACIs for authentication of those keys, SelfCertify.com can provide standardized digital certificates (e.g., using the X.509 standard) to ensure that its subscriber's public keys can be validated in a manner compatible with conventional public key infrastructure. Again, SelfCertify.com does not pretend that the trust imparted by its digital certificates is based on its confirmation of the identity of its subscribers. Instead, SelfCertify.com makes a policy of only issuing certificates for public keys that subscribers have self-certified with their notarized ink signatures in ACI documents. By signing a public key with its X.509 certificate, SelfCertify.com simply indicates that it has reviewed the original ink ACI and that a copy of the document can be freely downloaded from its Web server. | C-4, para. 3 |
| **0063** | The use of X.509 or other standard certificates permits SelfCertify.com to live in the world of conventional CAs even though it is based on an entirely different trust model. Users who accept the covenant trust model can install SelfCertify.com's root CA certificate (the "grandfather" certificate that validates all of its individual certificates) into their Web browsers and e-mail applications. As the covenant trust model gains acceptance in E-commerce, the manufacturers of Netscape Navigator and Internet Explorer can be expected to incorporate SelfCertify.com's root CA certificate into their browsers, alongside the certificates of VeriSign, entrust, and dozens of other CAs. Subscribers who use PGP (Pretty Good Privacy) and are looking for a way to validate their public keys outside PGP's "web of trust" model can submit their public keys to SelfCertify.com for it to be signed by SelfCertify.com's own PGP signature. | C-4, para. 4;<br>C-5, para. 1 |
| **0064** | Because covenant trust does not require a trusted third party, subscribers' public keys can be validated directly from the subscriber's ACI. The public key of a SelfCertify.com subscriber can be validated by freely downloading a copy of the subscriber's ACI and checking its positive identification of the public key. Thus, no CA certificate is required at all. In fact, subscribers can directly distribute copies of their ACI to anyone who will be relying on signatures corresponding to their public keys. | C-5, para. 2 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| **0065** | Example Transaction Using SelfCertify.com | C-5, Subheading II |
| **0066** | Below is a brief description of an example transaction based on covenant trust. In this example transaction, SelfCertify.com serves as a third party for the following: (1) freely distributing a compact cryptographic software module to signer and recipient with instructions for secure use~~. The~~ -- the parties use the software for generation of the signer's private/public key pair, generation of the signer's digital signature on an electronic record, and validation of the digital signature against the signer's public key; (2) accepting credit card payment (with SSL encryption), public key codes, and full legal names of new subscribers to SelfCertify.com; (3) issuing blank ACIs to new subscribers, upon payment, with instructions for use; (4) scanning original signed ACIs received from new subscribers and posting digitally certified copies on the web for free downloading; and (5) retaining original ACIs in a vault for inspection by experts, judges, juries, or attorneys during dispute resolution. | C-5, para. 3 plus numbered paras. 1-5 following *(formatting changes from numbered paras. to inline parenthetical numbering not shown)* |
| **0067** | For convenience, this example refers to a widget vendor named Alice and a purchaser named Bob. (These names seem to be used in just about every published example of cryptographic transactions.) Alice wishes to sign a purchase agreement acknowledging Bob's payment of $1[M],000,000 for 10,000 widgets and promises to deliver the widgets immediately. Bob wants to make sure that Alice, the president of Widgets Inc., is the person signing the agreement and not some "man-in-the-middle" impostor. | C-5, last para. |
| **0068** | Signer Enrollment | C-6, bulleted subheading |
| **0069** | Alice visits SelfCertify.com and quickly downloads a copy of "SelfCertify," a simple, compact, secure, and free cryptographic software application for the Microsoft Windows 98/NT/2000, operating system with versions available for various other operating systems. The SelfCertify software installs to the Windows tray as an icon, with various functions selectable by right-clicking on the icon. If she wishes to avoid the need for installation, Alice has the option of simply downloading a single executable file to her desktop and running it from there. For maximum convenience (but possibly less security), a ~~Java~~ JAVA version of the software can be offered for execution in a web browser. | C-6, para. 1 |
| **0070** | Because SelfCertify.com serves its pages under SSL with a certificate issued by a conventional CA, Alice is assured that the software is authentic and trustworthy. For additional assurance, Alice reviews | C-6, para. 1, cont'd |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | statements on the security of the software, written and digitally signed by various cryptographic experts, and validates the signatures of the statements before relying on the software. | |
| **0071** | Alice then follows the procedures outlined on SelfCertify.com for generating a public key from a secure passphrase. (See Appendix X of the application as originally filed and, the discussion below of a "pronounceable passphrase worksheet".) She then gets out her credit card and subscribes to SelfCertify.com with her credit card number, public key code, and full legal name. | C-6, para. 2 |
| **0072** | SelfCertify.com then issues Alice a custom-generated PDF file, from which Alice obtains two printed pages. The first page is a blank ACI with a space for her driver's license or other photographic ID and the second page is customized security paper with Alice's key code printed repeatedly in the background in an outline font. | C-6, para. 3 |
| **0073** | Alice tapes her driver's license to the blank ACI in the space provided and places it on the glass of her photocopier, with the security paper at the top of her photocopier's paper supply. She then photocopies the blank ACI to produce an ACI, ready for her signature, with outline digits of her key code throughout its background. | C-6, para. 4 |
| **0074** | Alice then checks the key code against her public key to make sure it is accurate, goes to the Notary Public down the hall, and executes the ACI in the presence of the notary. The notary examines Alice's driver's license, notes (in the ACI) any security features of it such as a hologram or colored background lines, and signs and stamps the ACI. Alice has now entered into a legally binding covenant with any person bearing the ACI or a facsimile copy of it. (So that she can keep a copy for her files and make certified copies herself, Alice elects to prepare and execute two original copies of the same ACI before the notary.) | C-6, last para. |
| **0075** | Alice mails the executed ACI to SelfCertify.com. Within a few days, SelfCertify.com scans the ACI and posts a copy of it on its web site in PDF or TIFF format. SelfCertify.com stores the original ACI in a vault for possible inspection in the future by experts, judges, juries, or attorneys during dispute resolution. SelfCertify.com then emails Alice the following message: "Your Authorization and Certification Instrument (ACI) has been recorded and you are now listed as a fully enrolled subscriber of SelfCertify.com with key ABC01. Once you enter the enrollment password ["]'3f8u2b'["] in your SelfCertify software, your software will automatically download the latest copy of our public key registry (now including your key) and will | C-7, para. 1 plus following block quote |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | automatically validate your digital signatures with the following text in any messages you sign: ["]'The following text has been signed with a public key registered as key ABC01 at SelfCertify.com. Alice B. Costas has signed a written covenant not to repudiate digital signatures created with this public key. To view a copy of this document, click here. The code of this public key is BD7D F2FD EC1C DF14 4811 574F F7CE 7D1E 6EB6 F7E9 CCF7 208B.]"]' Persons relying on your digital signature will be able to easily download and inspect a copy of your ACI to legally bind you to that signature." | |
| **0076** | Signer's Digital Signature of Electronic Record | C-7, bulleted subheading |
| **0077** | In her email software, Alice selects the text of her purchase agreement with Bob and right-clicks on the SelfCertify icon in the Windows tray. She then selects the menu item "sign" and, when prompted, enters her private key passphrase. She will probably have to look the passphrase up from a piece of paper in her purse the first few times she uses it. Later, she will put the piece of paper in her safe or destroy it if she trusts her memory enough. If she forgets or loses the passphrase, it's not a big deal. She only needs to create another public key from a new passphrase, cancel her original ACI, and request another one to continue signing records. | C-7, para. 1 after bulleted subheading |
| **0078** | As soon as Alice has entered her passphrase, the text she selected in her HTML-formatted email is replaced by text that is identical (including any formatting) except for a block of hexadecimal codes and the following statement in a reduced-size font: "I, Alice B. Costas, have signed this document with my public key, which is registered as key ABC01 at SelfCertify.com. To verify this signature, click on http[://] (colon, double forward slash) SelfCertify.com/validate to download a compact, virus-free signature verification program that confirms the signature and public key. The software will allow you to obtain a copy of a paper document that you can use to legally bind me to this digital signature. You can also independently validate the public key by clicking on http[://] (colon, double forward slash) SelfCertify.com/?ABC001 to view a digitally certified copy of the document." | C-7, para. 2 after bulleted subheading plus following block quote |
| **0079** | The formatting of the original text is preserved in the signed version. There is no header to the block of signed text because the SelfCertify software automatically calculates the beginning of the signed text block based on the number of signed characters, which is recorded in the signature block. Alice is free to select only a portion of the text for signature. For example, she may choose not to include letterhead | C-7, last para.;<br>C-8, para .1 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | at the top of her letters in the block of text she signs. | |
| **0080** | Alice can also use S/MIME email software such as Netscape Messenger or Outlook Express to sign email messages using conventional, standardized digital signature technology and the Covenant Trust model, without the need for the SelfCertify.com software. However, she needs to sign an ACI with the SHA1 fingerprint of her S/MIME public key (called a "Digital ID") to authenticate it under the Covenant Trust model. SelfCertify.com then can issue a certificate for her S/MIME public key to authenticate it, based on her ACI. | C-8, para. 2 |
| **0081** | Recipient's Validation of Digital Signature | C-8, bulleted subheading |
| **0082** | Bob receives Alice's digitally signed purchase agreement and downloads the SelfCertify software from the link provided in Alice's signature block. He also downloads a copy of her ACI. Once the software has been installed as an icon, Bob selects Alice's entire e-mail and right-clicks on the icon, then selects "Verify." A window pops up that says: "The following text has been signed with a public key registered as key ABC01 at SelfCertify.com. Alice B. Costas has signed a written covenant not to repudiate digital signatures created with this public key. To view this paper, click here. The code of this public key is BD7D F2FD EC1C DF14 4811 574F F7CE 7D1E 6EB6 F7E9 CCF7 208B." | C-8, para. 3 plus following block quote |
| **0083** | Since this is a $1[M],000,000 deal and he has never used the software before, Bob is not content with the software's assertion that Alice has entered into a legally binding covenant not to repudiate her digital signature with this key. Plus, Bob wants to have his lawyer look over the language of the covenant. So he clicks on the "here" link and a viewer window pops up with a TIFF copy of Alice's ACI. He prints out the ACI, notes that Alice's signature (which he recognizes from previous paper-based contracts) has been notarized and that the key code in the ACI is reproduced throughout the background of the document as vertically oriented digits in various outline fonts. The digits intermingle with the signatures, notary stamp, handwritten annotations, and images from Alice's driver's license. The key code digits even show up in the background of Alice's photograph in her driver's license. | C-8, second to last para. |
| **0084** | Bob needs no further convincing that Alice was the one who signed purchase agreement. His lawyer, however, wants him to check out SelfCertify.com's SSL certificate for the copied ACI. Bob downloads the ACI copy from SelfCertify.com and, with the image of the ACI in | C-8, last para.;<br>C-9, para. 1 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
|  | his Web browser, clicks on the "security" button of the browser. The browser provides a certificate issued to SelfCertify.com from a major CA, and Bob's lawyer is satisfied. |  |
| **0085** | If Alice uses S/MIME to digitally signed her message, Bob can simply trust her S/MIME "Digital ID" based on the certificate SelfCertify has issued for it. Thus Alice and Bob can use the Direct Trust model with S/MIME signatures and conventional digital certificates, trusting SelfCertify.com as a CA only for inspecting and verifying Alice's CA against the standard covenant language of the ACI, which is published at SelfCertify.com. | C-9, para. 2 |
| **0086** | Alternatively, Bob can download and review Alice's ACI for her "Digital ID" from the web site of SelfCertify.com. If Bob chooses to download Alice's ACI, he will need to open Alice's "Digital ID," look for her SHA1 fingerprint, and compare it to the fingerprint printed on her ACI. This alternative procedure, while requiring an extra step, provides S/MIME signatures based more directly on the Covenant Trust model, moving closer to the ultimate link in the chain of authenticity, which is the original notarized, ink-signed ACI paper. | C-9, para. 3 |
| **0087** | The key code in the ACI can be printed throughout the background of the entire paper as vertically oriented digits in various outline fonts, as in the example ACI discussed above. The font types, sizes, spacings, and line spacings are varied pseudorandomly in each ACI to make it difficult for an attacker to create an identical field of digits, which the attacker could use to remove the digits (by an XOR operation) from the ACI and substitute his or her own digits. Every bit of text and authenticating indicia in the ACI has background digits running through it. This feature (and possibly other features such as varying the spacing between digits of the text in a coded manner) protects both the signer of the ACI and the person relying on the ACI. | C-9, first bulleted para. |
| **0088** | The ACI can be created with a two-step procedure using a first page that is a blank ACI with a space for her driver's license or other photographic ID and a second page that is customized security paper with the subscriber's public key code printed repeatedly in the background in an outline font. The subscriber tapes his or her photo ID to the blank ACI in the space provided in places it on the glass of her photocopier, with the security paper at the top of the photocopier's paper supply. The blank ACI is then photocopied to produce an ACI, ready for the subscriber's signature, with outline digits of her key codes throughout its background. | C-9, second bulleted para., cont'd on C-10 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| **0089** | The ACI can include language that makes it the only printed document of its type that can be accepted as valid. Additional ACIs can be signed electronically for additional keys, but they must be signed with the key that is certified in the original paper ACI. SelfCertify.com attaches digitally signed ACIs (for a fee) to the PDF or TIFF file in which it distributes the original paper ACI. By ensuring that the original printed document disclaims all other documents purporting to bear the singer's handwritten signature, a "strength in numbers" validity system is established that gives the authenticity of a widely distributed ACI, publicly available from a trusted server, far more weight than a single forged copy having a different key code. This feature helps to protect the signer of the ACI. | C-10, first full bulleted para. |
| **0090** | The SelfCertify software can employ an ECDSA public key signature system with NIST Elliptic Curve P-192 (equivalent to 80-bit key length of symmetric cipher). The elliptic curve is described by a GF(p) field, where p is prime, to avoid recent attacks on elliptic curves from $GF(2^m)$, where m is a composite of smaller primes. See Smart, N. et al., "Constructive and Destructive Facets of Weil Descent on Elliptic Curves," HP Technical Report HPL-2000-10, 17 January 2000.) A 192-bit public key can be represented by 12 groups of 4 hexadecimal digits. The short key length made possible by elliptic curve cryptography makes it easy for a recipient to visually verify the entire key code against the printed text of an ACI and the background security digits. | C-10, second full bulleted para. |
| **0091** | The subscriber can be instructed to use a standardized, pronounceable passphrase made of "pseudowords" with alternating consonants and vowels. The passphrase is designed to be relatively easy to memorize, pronounce, and type and is very secure, with an entropy of about $2^{64}$. The passphrase is created with simple, secure system using a piece of paper and a paper clip for random selection of digits. | C-10, third full bulleted para. |
| **0092** | An SHA-1 hash of the passphrase can be used as the private key, with the subscriber's full legal name (from the SelfCertify.com directory) incorporated (transparently to the signer) into the passphrase as "salt." The use of salt prevents passphrase attacks using precomputed hashes of passphrases within the standardized approximately $2^{64}$ passphrase space. | C-11, fourth full bulleted para. |
| **0093** | Formatting of signed text can be preserved after signing. The added text of the signature block is formatted in an unobtrusive font that does not detract from the appearance of the signed text. The text in the signature block includes a data field with the number of | C-10, last bulleted para., cont'd on C-11 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | characters being signed, which avoids the need for a distracting header block (e.g., "-----BEGIN PGP SIGNED MESSAGE-----" in PGP). Documents can also be signed as files, in which case the signature resides in a separate ".SIG" file, as is conventional. | |
| **0094** | ACIs can be automatically opened from the software's signature validation window, based on the identification information in the signature block, and displayed or printed from a compact viewing window. | C-11, last para. |
| **0095** | ~~See~~ FIG. 1 is a flow diagram of a signer enrollment procedure of an exemplary self-certification process, | P3/L6; B-1 *("SIGNER ENROLLMENT")* |
| ... | involving a signer 120 and a recipient 150. | B-1 *(FIG. 1, refs. 120, 150)* |
| ... | The process includes several transmissions to and from the SelfCertify.com site, block 110, | B-1 *(FIG. 1, seven arrows total entering and leaving block 110)* |
| ... | discussed ~~See flow diagram of an exemplary process in Appendix B and A description of the method, in Appendix C, Appendix C is adapted from a confidential document describing benefits of the inventive method (as a technology rather than as a product or service) generally and~~ in the conjectur[e]al ~~only~~ example above. | P3/L6-9 |
| ... | The site serves its pages under SSL (a conventional certificate authority 112, *e.g.*, Thawte, connects to site 110). | B-1 *(FIG. 1, refs. 134, 144; block 112 → block 110)* |
| **0096** | Signer 120 downloads signature software 122 from SelfCertify.com at 121 | B-1 *(FIG. 1, block 110→refs. 121, 122)* |
| ... | and then employs a passphrase 124 to ~~signer~~ create[s] a public key at 123. | B-1 *(FIG. 1, ref. 123→ref. 121, ref. 123→ref. 124)* |
| ... | Signer 120 submits the public key at 125. ~~S.C.~~ The site, at 126, issues an auth[.]entication document for signer 120 to print and sign. | B-1 *(FIG. 1, refs. 125, 126)* |
| **0097** | At 131, signer 120 executes the auth[.]entication doc[.]ument. At 132, signer 120 mails the document to S[.]elfC[.]ertify.com. | B-1 *(FIG. 1, ref. 131→ref. 132)* |
| **0098** | At 133, S[.]elfC[.]ertify.com reviews, stamps, and scans the document to a PDF or TIFF file 142. | B-1 *(FIG. 1, ref. 132→ ref. 133)* |
| ... | ~~PDF TIFF~~ File 142 is then available along with the site's SSL | B-1 *(FIG. 1, refs.* |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | certificate 144 and a PGP signature 146, | *144, 142, 146 grouped together by brace to their right)* |
| ... | which results from PGP signing at 135. | B-1 *(FIG. 1, ref. 135 → ref. 146)* |
| **0099** | FIG. 2 illustrates a procedure 200 for signing [+] and verif[y]ication | B-2 *("SIGN + VERIFY")* |
| ... | in the exemplary self-certification process. | B-2 *("STEP 2")* |
| ... | Signer 120, using passphrase 213, signs message 212 at 214, | B-2 *(FIG. 2, refs. 212-214)* |
| ... | which is combined at 217 with ~~signer sends~~ instructions 215 sent by signer 120 at 216. | B-2 *(FIG. 2, refs. 214, 216 → ref. 217)* |
| ... | At 222, recipient 150 obtains the signed message w[/]ith instructions 217 and follows the instructions. [#1 -] First, recipient 150 downloads "PEGWIT" w[/]ith the latest signed public key file. | B-2 *(FIG. 2, ref. 217 → ref. 222)* |
| ... | At 224, optionally because the keyfile is signed by SelfCertify.com, e.g., [(]w[/]ith PEGWIT[?)] or just SSL[)] (FIG. 1, 134), the ~~downloads~~ auth[.]entication doc[.]ument for signer 120 is downloaded. | B-2 *(FIG. 2, ref. 224 and its NOTE 1)*; B-1 *(FIG. 1, ref. 134)* |
| ... | At 226, the ~~verify~~ message is verified. | B-2 *(FIG. 2, ref. 226)* |
| **0100** | [3.] An alternative method of authenticating a subscriber compris[ing:]es having the subscriber agree to terms of the ACI (directly or by reference to an ACI displayed on a secure web page) in a recorded telephone conversation. ~~See Appendix D for an example.~~ See Appendix E of the application as originally filed for a disclosure of a technique  for authenticating the recording. | P3/L13-16 |
| **0101** | One advantage of this method over the use of a holographically-signed paper ACI is that a person relying on the subscriber's digital signature is likely to know the sound of the subscriber's voice, and is thus likely to trust a recorded verbal agreement. SelfCertify.com can sign a digital file containing the recording (e.g., in compressed WAV format) as it can a PDF/TIFF file of a paper ACI. A recorded "verbal ACI" can have security features somewhat analogous to those of a paper document. For example, a synthesized voice can recite digits of the fingerprint of a public key certified by the ACI repeatedly throughout the entire recording, preferably with subdued volume. | P3/L17-24 |
| **0102** | ~~3. Method of authenticating a subscriber comprising: having the~~ | P3/L13-15 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
|  | ~~subscriber agree to terms of the ACI (directly or by reference to a ACI displayed on a secure web page) in a recorded telephone conversation. See Appendix D for~~ The following is an example of subscriber authentication by a recorded telephone conversation. |  |
| **0103** | Alice wishes to become a subscriber to SelfCertify.com so that Bob will rely on her public key. However, she doesn't wish to go through the hassle of having a paper document sent to her and having it signed in the presence of a notary. She also wants people to be able to authenticate her public key by hearing a simple recorded statement by her. So, she chooses the "Verbal ACI" option on the SelfCertify.com web site and enters her phone number and the fingerprint of her public key into the form. The web site then lists a phone number and an access code and incites her to call the number. | D-1, para. 1 |
| **0104** | She dials the number (making sure that the call blocking is disabled so that SelfCertify.com can detect the phone number she's calling from) and enters the access code using the touchtone keys of her telephone. She then enters into a brief oral exchange with a computer or human operator at SelfCertify.com. The exchange goes something like the following in TABLE I below: | D-1, para. 2 |
| **0105** | TABLE I *(Table contents omitted. Blanks indicated with underlining in source material changed to "(blank)" in table contents.)* | D-1 |
| **0106** | ~~<End of~~ The recording[>] then ends. | D-1, last line |
| **0107** | A cryptographic system usability enhancement[s -] ~~usability~~ of the inventions is [19.] signing an electronic record (e.g., MS WORD document, AUTOCAD drawing etc.) by "printing" that record to a special "virtual document printer." ~~See Appendix AD.~~ The "virtual signature printer" provides the user with an intuitive, simple way of authenticating an electronic record. | P12, heading plus following paragraph |
| **0108** | An example of a virtual printer is the "PDF Writer" printer driver that is installed with the ADOBE ACROBAT software. The inventive "virtual signature printer" provides the user with an intuitive, simple way of authenticating an electronic record. | AD-1, para. 1 |
| **0109** | The "virtual signature printer" system produces an output file (or multiple files, see below) that can be sent to a recipient for viewing, printing, and validation. The recipient can view or print the file (preferably, the file is ["]backward compatible["] ~~compatible~~ with widely available viewing software) and, with special software, can validate the signature on the file. | AD-1, para. 2 |
| **0110** | A particularly advantageous way of signing an electronic record that | AD-1, para. 3 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
|  | has been "printed" this way is with embedded signatures. However, a "virtual signature printer" system can employ any suitable technique for signing an electronic record. The following are some examples of output of such a system: (1) a PS or TIFF file (or, with suitable licensing if necessary, a PDF file) representing the document, accompanied by a detached PGP-compatible signature file[.]; (2) a ZIP file containing a PS, TIFF, or PDF file representing the document including a ZIP comment containing a Base-64 PGP-compatible signature[.]; (3) a PGP-signed file containing the document in a PS, TIFF, or PDF file. | plus following bulleted paragraphs |
| **0111** | When the signer wishes to electronically sign a document he or she can print the document to the virtual signature printer driver, using the print functionality of the software used to create the document. The printer driver creates a window in which the software requests the signer's authenticating information. The user can enter his or her passphrase, apply his or her fingerprint to a fingerprint scanner, insert a smart card, etc. The software then computes the digital signature for the document, based on the authenticating information or a private key unlocked by the authenticating information, and embeds the digital signature with an output file or includes the digital signature in a separate file. | AD-1, last para. |
| **0112** | ~~The following terms are said to be elements of the inventions because, in various combinations (and in some cases standing alone), they recite subject matter that the applicant views as particularly pointing out various aspects of his inventions.~~ The terms "virtual signature printing," "virtual signature printer" are to be broadly understood as including any mathematical construct, structure, method, system, etc., as the case may be, suitable for carrying out the function~~(s) discussed~~ of | P18/L5-10 |
| **...** | ~~"Virtual signature printing," "Virtual signature printer."~~ authenticating an electronic record such as a word processor document by "printing" the document using a printer driver that does not actually produce printed output, al least not as its main purpose. Instead, such a printer driver according to various aspects of the inventions creates another file that includes, or references, indicia of the user's digital signature authentication of the electronic record. | P23/L11-16 *(one entry in the list of elements of the inventions referred to at P18/L5-10)* |
| **0113** | "Print to Signed TIFF File" Process | AF-1, heading |
| **0114** | According to various aspects of the present inventions, a document can be signed by printing the document to a TIFF file using a virtual printer driver, e.g. provided by a service such as "SelfCertify.com." | AF-1, para. 1 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | (Note that SelfCertify.com is not an operating business entity, though the applicant has registered the domain name, and has not offered any product or service for sale as of the filing date of the present provisional application.) The TIFF file is created as it normally would be except that includes a signature block within a suitable field. | |
| **0115** | In one embodiment, the signature block is included within the TIFF "ImageDescription" field, the ASCII contents of which are excluded from the signature calculation of the file. ~~(~~See ~~AF-3.)~~ the following table for a | AF-1, para. 2 |
| ... | sample "ImageDescription" TIFF Field: | AF-3, heading |
| **0116** | TABLE II *(Table contents omitted.)* | AF-3 |
| **0117** | In an actual implementation, the user "prints" the document to the digital signature printer driver. The driver will creates the TIFF (multi-page if necessary), with blank characters in the fixed-length "ImageDescription" field where the signature data is intended to reside (setting that data to a default blank ASCII value), compute signature for the entire document with the default blank value in the "ImageDescription" field, and place the signature data in the "ImageDescription" field as ASCII. | AF-1, para. 2 |
| **0118** | The document's signature can be verified simply by opening it with a customized TIFF reader, which extracts the signature data from the "ImageDescription" field and validate it against the data of the document with default blank values substituted in the "ImageDescription" field. | AF-1, para. 3 |
| **0119** | An exemplary process 300 is illustrated in ~~the attached~~ FIG. 3 ~~(AF-4)~~, in which a signer creates a document 310 using whatever software he (or she) wishes to use (e.g., MICROSOFT WORD 97). | AF-1, para. 4 |
| ... | At 312, he or she then prints ~~the~~ document 310 to the SelfCertify.com virtual TIFF printer, optionally with ACI 311. | AF-1, para. 4;<br>AF-4 *(FIG. 3, arrow from ref. 311 to ref. 312 marked "option")* |
| ... | At 314, ~~compute~~ a signature for the resulting TIFF file is computed, with blanks in the TIFF "ImageDescription" field. | AF-4 *(FIG. 3, ref. 312 → ref. 314)* |
| ... | At 316, the ~~replace~~ blank characters are replaced with the signature data. The result is a multi-page TIFF file 318 with a "comment" of the signature block. | AF-4 *(FIG. 3, ref. 316 → ref. 318)* |
| **0120** | ~~He or she then prints the document to the SelfCertify.com virtual~~ | AF-1, para. 4; |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | ~~TIFF printer.~~ The printer driver software ~~creates a~~ of the virtual TIFF printer displays the TIFF file 318 of ~~the~~ document 310 ~~and displays it~~ in a viewer window. The user interface of the viewer window requests authentication from the signer to apply his digital signature to the document. | AF-4 *(FIG. 3, refs. 318, 310)* |
| ... | In exemplary process 300, a ~~compute~~ signature is computed at 314 based on a signing key 313. | AF-4 *(FIG. 3, ref. 313 → ref. 314)* |
| ... | The authentication can be a securely delayed passphrase input according to various aspects of the present inventions. | AF-1, para. 4 |
| **0121** | Advantageously, the signed document 318 is in a conventional format (e.g., multi-page TIFF) that can be read by any conventional viewer i[s]f signature authentication is not needed. When signature authentication is needed, a document can be viewed using the SelfCertify.com TIFF viewer, [(]which may be distributed freely to encourage use of SelfCertify.com's digital signature services[)]. | AF-1, para. 5 |
| **0122** | At 320 in process 300, a public key algorithm generates a public key 322, which is used for signature validation at 341 of path 340, as discussed below. | AF-4 *(FIG. 3, ref. 320 → ref. 322 → ref. 341 inside boundary 340)* |
| ... | The signer's public key 322 can be included in the signature data along with the digital signature and the name of the signer. A facsimile copy of the key ACI 311 (see, e.g., Appendix A of the application as originally filed) can be appended to the end of ~~the~~ TIFF file 318 after the document pages. | AF-1, last para. |
| **0123** | Having both of th[e]ose items present in the TIFF file permits a verifier to quickly authenticate the signed document. If he or she is satisfied with the integrity of the TIFF reader/verification software he or she has obtained (e.g., from a secure, trusted web page of SelfCertify.com), and if he or she is satisfied that the facsimile copy of the ACI with its security background digits is not a forgery, he will have everything he needs to validate the signature of a person who has not made any prior digit[i]al [as]signing arrangements with him. | AF-2, para. 1 excluding portion begun on AF-1 |
| **0124** | Signing and verification software according to various aspects of the present inventions can be integrated with the GPG ("GNU Privacy Guard") software, which can be freely distributed and modified under the GNU Public License. The signing and verification software can call the GPG software with parameter passing. The ~~inventor of~~ inventive signing and verification software can thus include, generally, a slightly modified TIFF printer driver and TIFF reader that calls the GPG software for all digital signature | AF-2, para. 2 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | functionality. All three pieces of software can be released in a single compact package. | |
| **0125** | The TIFF reader software can provide the option to output the original TIFF file (without signature data in the "ImageDescription" field) along with a PGP-compatible detached signature to the file and an ASCII file with the signer's public key. This permits less trusting users to verify integrity of the signature and signing key (comparing PGP's SHA-1 "fingerprint" to what's shown on the ACI) with their own copy of PGP or GPG. | AF-2, para. 3 |
| **0126** | In one path 330 of exemplary process 300, signed ~~multi-page~~ TIFF file ~~with "comment" of signature block~~ 318 is viewed as document image 350 ~~view TIFF~~ w[/]ith a conventional TIFF viewer, at 332. | AF-4 *(FIG. 3, ref. 318 → ref. 332 inside boundary 330 → ref. 350)* |
| ... | ~~Advantageously, the signed document is in a conventional~~ As discussed above, the TIFF format ~~(e.g., multi-page TIFF) that can be read by any conventional~~ of file 318 permits such view[er]ing ~~is signature authentication is not needed~~. | AF-1, para. 5 |
| ... | Validat[e]ion of the signature, on the other hand, occurs in path 340 of process 300 at 341, 342, 344. | AF-4 *(FIG. 3, refs. 341, 342, 344 inside boundary 340)* |
| **0127** | At 342, ~~extract~~ signature block data[,] is extracted from file 318 and replaced with blanks. | AF-4 *(FIG. 3, ref. 318 → ref. 342)* |
| ... | (An alternative[:] is to use a conventional TIFF [+] file with a detached PGP signature, [(]e.g., in a ZIP file.) | AE-5 *(note on top of page)* |
| ... | At 341, the ~~validate~~ signature is then validated against the modified data | AF-4 *(FIG. 3, ref. 341)* |
| ... | and the public key 322 generated from signing key 313. | AF-4 *(FIG. 3, ref. 313 → ref. 322 via ref. 320)* |
| ... | Validation output 346 of that act is used at 344 to add validation data to the image of file 318 | AF-4 *(FIG. 3, refs. 318, 346 → ref. 344)* |
| ... | before viewing it as document image 350. | AF-4 *(FIG. 3, ref. 350 is result of "view[ing]" at ref. 332 & can also be result of ref. 344)* |
| **0128** | ~~Can use~~ Existing technology can be used to print and sign documents according to various aspects of the present inventions. | AE-5 *(note at lower right of page)* |

| ***Para.*** | ***Substitute Specification Text with Changes from Source Material*** | ***Source Material*** |
|---|---|---|
| ... | For example, the "PEERNET.DRV TIFF Driver converts any document capable of being printed by a windows application into high quality serialized or multi-page TIFF images. It is ideal for imaging or archiving applications. It's also a handy file-generation tool for cross-platform article distribution. TIFF conversion is as fast as printing. Document scanning is obsolete. Paper waste is a thing of the past" (Appendix AE-5 of the application as originally filed). | AE-5 *(block of text indicated by arrow leading from note referenced above)* |
| **0129** | Another example of existing TIFF generation technology that may be adapted for use according to various aspects of the present inventions | AE-6 *(note at upper right of page)* |
| ... | is the "EZ-Printer for Windows NT," which "makes it possible to print from any application to an image file. It installs itself as a native device in the printer control panel. Users can simply choose Print and select the EZ-Printer printer from the list of available printers" (Appendix AE-6). | AE-6 *(description)* |
| **0130** | Other cryptographic system usability enhancements [-] ~~usability~~ are | P12/L2 |
| ... | an embedded signature[:], where graphical indicia of a signature is placed within a graphical (image-based) representation of an electronic record, | P12/L7-8 |
| ... | and preserving formatting (e.g., font size, boldface, underlining, etc.) of clear-signed text. | P12/L16 |
| ... | ~~This brief disclosure is an~~ For example, ~~of~~ a clear-signed document ~~with~~ can have a simulated digital signature (25x13 bits = 325 bits) that resides as a graphic within a signature block that is excluded from the signature calculation of the document. In an actual implementation, the user will place the signature block in the document and will then "print" the document to the digital signature printer driver. The driver will then create a graphic file such as a TIFF (multi-page if necessary), remove the graphics in the region where the signature block will go (setting that graphic data to a default blank value), compute signature for the entire document exempt the signature region, and place the signature data in the region as graphic-mode text. | AE-1, para. 1 of letter body |
| **0131** | The document's signature can be verified simply by opening it with a customized TIFF reader, which will detect the presence of the signature region and will validate the signature within it against the data of the document except the graphics within th signature block. An option can be provided to put the signature data on and entirely separate page of the document (e.g., after the last page), preferably with a facsimile copy of the signer's ACI. (In such embodiments, the | AE-1, para. 2 of letter body |

| ***Para.*** | ***Substitute Specification Text with Changes from Source Material*** | ***Source Material*** |
|---|---|---|
|  | ACI should have a blank space for the signature data of document signed with the ACI's signing key.) |  |
| **0132** | An exemplary process 400 is illustrated in ~~the attached~~ FIG. 4, in which a signer creates a document 410 (e.g., ~~this~~ a letter) using whatever software he (or she) wishes to use (e.g., MICROSOFT WORD 97). ~~He then~~ At 412, the signer prints the document to the SelfCertify.com virtual TIFF printer, which acts as ~~(call it~~ as a "software signature machine."[?)] The printer driver software creates a TIFF file of the document and displays it in a viewer window. The user interface of the viewer window requests ~~the following input from~~ that the signer selec~~tion of~~ a graphical region within the displayed document for application of the signer's digital signature "stamp." | AE-1, last 2 paras. of letter body |
| **0133** | The user can specify the region by moving a dashed box around the screen, as illustrated in FIG. 5. ~~The~~ Dashed box 510 appears over text of a document to be signed 500. Dashed box 510 includes left and right arrows 512, 514 within it for navigating to different pages of a multi-page document, and a[n] "sign here" (or "OK") button 516 for applying the digital signature "stamp" at the current location of the box. ~~An unsigned signature page of this letter is attached~~ FIG. 5 show[ing]s what ~~such a~~ box 516 might look like as it is moved around during the selection process. | AE-1, last para. of letter body; AE-2, para. 1; AE-3 *(dashed box appears over text of letter)* |
| **0134** | The user interface further requests authentication from the signer to apply his digital signature to the document within the digital signature "stamp" at the selected location. As discussed ~~in other disclosures~~ below, the authentication can be a securely delayed passphrase input. | AE-2, para. 2 |
| **0135** | FIG. 4 illustrates an exemplary process 400 ~~is illustrated in the attached FIG. (AE 4), in which a signer creates~~ for digitally signing a document 410 | AE-1, para. 3, lines 1-2 |
| ... | ~~He then prints the document to the SelfCertify.com~~ by virtual print[er]ing | AE-1, para. 3, lines 3-4 |
| ... | to a TIFF file with a graphical signature block 418. | AE-4 *(FIG. 4, ref. 418)* |
| ... | When the signer has selected the location of the digital signature stamp and provided authentication for its creation (a signing key 413 is employed in exemplary process 400), the printer driver software performs operations at 414 (FIG. 4). It removes all graphic information from the selected location~~. It~~ and then computes the digital signature based on (1) the remaining data of the TIFF file | AE-2, para. 3; AE-4 *(FIG. 4, ref. 414: "compute signature for all data except sig. block region")* |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | (exclusive of the location of the stamp) and (2) the signer's private key. | |
| **0136** | ~~This brief disclosure is an example of a clear signed~~ FIG. 6 illustrates document 500 with a simulated digital signature ~~(25x13 bits = 325 bits)~~ that ~~resides~~ has been added at 416 as a graphic 600 within a signature block. | AE-1, para. 1 of letter body;<br>AE-4 *(FIG. 4, ref. 416)* |
| ... | Signature data of graphic 600 is in the form of groups of four decimal digits. | AE-4 *(FIG. 4, ref. 416)* |
| ... | Alternatively, ~~NOTE:~~ signature data can be in the form of a bar code (1D or 2D) and could include signature characters as data (not image pixels), making search a very simple search for key tag characters in the file. | AE-2 *(note at lower left of page)* |
| **0137** | Advantageously, the signed document 418 is in a conventional format (e.g., multi-page TIFF) that can be read by any conventional viewer, at 432 in path 430 of process 400 (FIG. 4), i[s]f signature authentication is not needed. When signature authentication is needed, a document can be viewed using the SelfCertify.com TIFF viewer (which may be distributed freely to encourage use of SelfCertify.com's digital signature services) in path 440 of process 400. | AE-2, para. 4;<br>AE-4 *(FIG. 4, ref. 432 inside boundary 430: "view TIFF w/ conventional viewer")* |
| **0138** | At 442, ~~render~~ graphics are rendered from TIFF file 418 and provided to 444 as well as data output 450. | AE-4 *(FIG. 4, ref. 418 → ref. 442)* |
| ... | At 444, ~~extract~~ signature block graphic 600 (FIG. 6) is extracted. OCR is performed on the result, at 445, and the OCR result is used to validate the signature of document 410 against remaining data of file 418, at 446. | AE-4 *(FIG. 4, ref. 442 → ref. 445 → ref. 446)* |
| ... | Path 440 of process 400 provides validation output 447 in addition to data output 450. | *AE-4 (FIG. 4, items in boundary 440 → ref. 447 & ref. 450)* |
| **0139** | To verify [a] the digital signature of file 418, the viewer first searches the graphical rendering of the signed document provided at 442 for the distinctive graphical outline of the signature block, at 444. | AE-2, para. 5 |
| ... | Distinctive features of the graphical outline can include (1) a distinctive color such as maroon, (2) a distinctive line shape such as double parallel lines, and (3) a distinctive line weight such as 3.2 points (not an integer or x 0.5 fraction). All of th[e]ose distinctive properties are present in ~~the~~ simulated signature block 600 for ~~this~~ document 500 (FIG. 6). In addition, the signature block can have a | AE-2, para. 5, cont'd; AE-2 *(FIG. 6, refs. 500, 600)* |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | predetermined size to further identify it. | |
| **0140** | Once the viewer software has identified the exact location of the signature block, it removes all of the graphical data of the signature block (up to and including the border) and sets it to the default blank value that was used during the signature calculation performed at 414. At 445, it performs an optical character recognition of the signature data within the block to obtain the value of the digital signature. According to advantageous aspects of the invention disclosed ~~elsewhere~~ below, the digital signature can be applied to a secure delay to obtain another, larger digital signature value. | AE-2, last para. of letter body; AE-4 *(FIG. 4, ref. 414: "compute signature for all data except sig. block region")* |
| ... | The digital signature value (as shown in FIG. 6 or as expanded through a secure delay) is then validated against the modified TIFF representation of the document. | AE-2, last para. of letter body, cont'd; AE-2 *(FIG. 6, ref. 600)* |
| **0141** | ~~25.~~ Other aspects of the invention are preserving formatting (e.g., font size, boldface, underlining, etc.) of clear-signed text~~, See Appendix AH.~~ | P12/L16-17; P2/L2 *(aspects vs. numbered paras.)* |
| ... | ~~26.~~ and employing an unobtrusive digital signature around the clear-signed text, ~~See Appendix AH~~ as illustrated in FIG. 7. | P13/L1-2; AH-1 *(FIG. 7)* |
| ... | A digital signature can be made unobtrusive in a number of different ways ~~including:~~. One way is to use a small, distinct font (e.g., 8 point Courier) for the Base-64 encoded signature characters, as in block 710 of clear-signed email 700. | P13/L3-4; AH-1 *(FIG. 7, ref. 710)* |
| **0142** | Another way of making a digital signature ~~can be made~~ unobtrusive ~~in a number of different ways including:~~ is to | P13/L1-2 |
| ... | omit[ting] the "header" that tells conventional digital signature software where to start looking for the beginning of clear-signed text. In the inventive alternative to such a header, the signature block (at the end of the clear-signed text) includes the number of characters in the clear-signed text, which allows the digital signature software to "count backwards" from the end of the clear-signed text until it reaches the beginning or that text. | P13/L5-10 |
| ... | Another way a digital signature can be made unobtrusive ~~in a number of different ways including:~~ is | P13/L1-2 |
| ... | using a white font color to hide the signature characters entirely. | P13/L11 |
| **0143** | Another cryptographic system usability enhancement[s -] ~~usability~~ | P12, heading |
| ... | is ~~23.~~ graphical (mouse) input of "CVCVCVCV" passphrase digits. | P12/L13 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | ~~See Appendix Z.~~ | |
| ... | ~~Z2-Z6~~ FIGS. 8-12 illustrate screen shots of a secure ~~passphrase entry~~ system according to various aspects of the invention for | Z-1, para. 1<br>*(FIGS. 8-12 are Z-2 through Z-6)* |
| ... | ~~Appendix Z: Discloses various aspects of secure passphrase entry according to the inventions, including~~ graphical entry of a "CV" passphrase[.], | P9/L14-15 |
| ... | illustrating an exemplary user interface at different points during the input of a passphrase without the use of keystrokes. Thus, the security hazard of keystroke loggers can be avoided. In addition, the mouse-based input method may be preferred by users over the use of a keyboard, for example when they are entering their passphrase to browse encrypted e-mails or files. In an experiment the applicant carried out, "entering" the passphrase by the mouse input methods (simulated by tapping a pen onto a printout similar to ~~Z2-Z6~~ FIGS. 8-12) did not take him much longer than typing in the passphrase. | Z-1, para. 1<br>*(FIGS. 8-12 are Z-2 through Z-6)* |
| **0144** | ~~11.~~ A "consonant-vowel" ("CV") passphrase of repeated pairs of consonants and vowels[.] | P9/L4-5 |
| ... | provides a user-friendly representation[s] of a high-entropy data word[s]. | P9/L3 |
| ... | The consonants are preferably selected to be phonetically and visually distinct. The set of consonants {b,d,g,h,k,l,m,n,p,r,s,t,z} has particular advantages. The use of "pseudowords" created from four adjacent CV pairs is particularly advantageous because the "pseudowords" are pronounceable and have a linguistic look and sound. | P9/L5-8 |
| **0145** | In FIG. 8, letters "NIH" have been entered (812) and mouse pointer 820 is near the letter "U" in a column 830 containing the letters AEIOU. | Z-2 *(FIG. 8, ref. 812: "NIH"; ref. 820: mouse pointer.; ref. 830: col. w/ AEIOU)* |
| ... | In the view of FIG. 9, the letter "U" is entered (914) where only an underscore 814 is present in the view of FIG. 8. | Z-3 *(FIG. 9, ref. 914: letter U)* |
| ... | A "private key delayed unlock" bar 840 has 2/17 of its length filled in. | Z-2 *(FIG. 8, two of 17 squares filled in)* |
| **0146** | In FIG. 9, letters "NIHUD" have been entered (912) and bar 840 is 4/17 filled in. | Z-3 *(FIG. 9, ref.. 912; 4 squares filled)* |
| ... | In FIG. 10, letters "NIHUDEZO PO" have been entered (1012) and | Z-4 *(FIG. 10, ref. 1012, ref. 820,* |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | mouse pointer 820 appears over the letter "Z" in a column containing the letters BDGHKLMNPRSTZ. Bar 840 is 9/17 filled in at this point. | *ref. 1030: col. w/ BDGH...RSTZ)* |
| ... | In FIG. 11, letters "NIHUDEZO POZOBUME" have been entered (1112), filling box 1113. Mouse pointer 820 is near the letter "E" in the right-most column 1130, and that letter appears (1114) as the last letter in box 1113. | Z-5 *(FIG. 11, ref. 1112, ref. 820)* |
| **0147** | Advantageously, the passphrase is represented in the illustrated embodiment (as it is entered) both as circled letters and [h]as a pair of stair-stepped line segments having characteristic shapes. | Z-1, para. 2 |
| ... | (FIG. 11 illustrates a stair-stepped line segment 1140 in one box 1150, connecting circles around letters 1141-1148, and another stair-stepped line segment 1160 in a box 1160 to the right of box 1150.) | Z-5 *(FIG. 11)* |
| ... | Viewing the passphrase and its associated characteristic shapes of the line segments helps the user to remember the passphrase. Human brains are good at remembering pronounceable words (even when they are nonsense words) and are also good at remembering characteristic shapes. The combination of both characteristics of a unique passphrase can be expected to improve the user's ability to remember it when the time comes to input the passphrase. | Z-1, para. 2, cont'd |
| **0148** | Another ~~24.~~ graphical (mouse) input usability enhancement is the entry of "Alphanumeric-Except-O" passphrase digits in a 7x5 grid[,]. ~~with already selected selected digits indicated. See Appendix AB.~~ | P12/L14-15;<br>P12, heading *("CRYPTOGRAPHIC SYSTEM ENHANCE-MENTS – USABILITY")* |
| ... | FIGS. 13-14 illustrate a particularly advantageous passphrase entry system using this "Alphanumeric-Less-O"passphrase type. | |
| **0149** | In FIG. 13, a box 1300 contains a grid 1320 with the letters A-N and P-Z, and all the decimal digits 1-9 and zero. ~~The~~ That arrangement is particularly advantageous because~~: (1)~~ the digits can be displayed in a 7x5 matrix (grid 1320)[,] and ~~(2)~~ confusion between the letter "O" and zero is avoided. | P9/L21-23;<br>AB-2, top half |
| ... | In a variation, an entry system is built into a keypad of a USB-connected hardware delay processor. Already-selected digits can be illuminated. | AB-2 *(note at right of page)* |
| **0150** | FIG. 13 illustrates a "passphrase:" box 1310 displaying an underscore but no entered digits. FIG. 14 illustrates box 1300 after entry of the digits "RJ27 X," which appear in box 1310 followed by an underscore. Those already-selected digits are indicated in grid 1320 by having been replaced with images depicting one, two, three, four, | P12/L14-15 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
|  | and five dots. |  |
| **0151** | FIGS. 13-14 are excerpted from Appendix AB of the application as originally filed. As illustrated in AB-1, an image depicting a single dot replaces the letter "R" in grid 1320 when that letter has been entered in box 1310. As illustrated in AB-2, an image depicting two dots replaces the letter "J"when that letter has been entered in box 1310, and an image depicting three dots replaces the decimal digit "2"when it has been entered in box 1310. As illustrated in AB-4, an image depicting four dots replaces the decimal digit "7"when it has been entered in box 1310. AB-4, AB-5, and AB-6 illustrate the replacement of "X," "Q," "H," and "3" by images depicting five, six, seven, and eight dots, respectively, when those letters and decimal digits have been entered. | AB-1; AB-2; AB-3; AB-4; AB-5; AB-6; |
| **0152** | An analysis of entropy using the preferred non-repeating ~~of~~ digitsin th[at]e system[,] of FIGS. 13-14 is found in Appendix AB. With ~~choices:~~ 7 x 5 = 35 = M choices and ~~digits:~~ N = 8 non-duplicate digits, the number of possibilities X = M•(M-1)•(M-2)•(M-3)...(M-N-), which works out to X = 6.67 x $10^{14}$. | P9/L25-26; AB-1 *(parts of top half of page)* |
| **...** | Based on the base 2 log of X, ~~log2(x)~~ [=] the result is 49.2 bits of entropy. With a 0.5 sec[.]ond delay between digits, d = 4 seconds, and 4X / (3600 sec./hr x 24 hr/d x 365 d/yr) = 84,496,818 years. | AB-1 *(parts of bottom half of page)* |
| **0153** | Returning to the ~~Z2-Z6 illustrate screen shots of a~~ secure passphrase entry system of FIGS. 8-12, | Z-1, para. 1 *(FIGS. 8-12 are Z-2 through Z-6)* |
| **...** | box 1113 is completely filled in FIG. 11 but "Private Key Delayed Unlock" bar 840 is 16/17 filled. | Z-5 *(FIG. 11, ref. 1113, ref. 840)* |
| **...** | In FIG. 12, bar 840 is completely filled and box 1113 displays "Passphrase confirmed. Your signature has been applied." | Z-6 *(FIG. 12, ref. 1113)* |
| **0154** | ~~16.~~ A secure delay[:] system 1500 with step-by-step delay processing~~.~~ ~~(See Appendix Z)~~ | P10/L15 |
| **...** | ~~A delay system~~ according to another aspect of the invention, illustrated in the block diagrams of ~~Z7 and Z8~~ FIGS. 15-16, makes a secure delay according to various aspects of the invention ~~less~~ unobtrusive to the user. It does so by beginning the delay process when the passphrase has been partially entered. | Z-1, para. 3 *(the word "less" clearly doesn't belong as the whole point of performing delay computations...* |
| **0155** | Advantageously, such a system performs the delay computations substantially in parallel with the unavoidable delay of the user's input of the passphrase. Even when typing quickly, it took the | Z-1, para. 3, cont'd. *(...substantially in* |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
|  | applicant at least about three seconds to enter the passphrase during ~~his~~ the experiment mentioned above. This is a substantial period of delay that, when made computationally unavoidable, makes cracking the $2^{48}$ possible combinations of the randomly chosen passphrase nearly impossible with the computing horsepower available around the date of filing of the present application. (See Appendix Z9 and Z10 of the application as originally filed for a detailed computational analysis.) | *parallel to unavoidable user delay is to make the overall delay more unobtrusive.* See *the following para. below)* |
| **0156** | By performing delay processing in steps and incrementally accepting user input (or displaying user output), the secure delay can provide the security benefit of increased intractability to an attacker without creating any noticeable inconvenience for the legitimate user. A secure-delayed passphrase entry system can process each keystroke (or graphical digit selection) input to produce succeeding delay output values, each of which is used as the initial value for the subsequent secure delay processing of the next input. A secure-delayed hash output system can display sub-hashes of an iteratively updated delay output value so that the user can visually compare digit clusters (e.g., "1234", "X7J", "zasadabi"~~, etc.~~) while the secure delay is running. | P10/L15-24 *(regarding deletion of the word "less"in para. above, note that "noticeable inconvenience for the legitimate user" is avoided, i.e., delay is made more unobtrusive)* |
| **0157** | The screen shots of ~~Z2-Z6~~ FIGS. 8-12 show the "private key delayed unlocking" beginning with the first consonant-vowel pair entered by the user. The delayed unlocking (the inventive "computationally unavoidable" delay) continues substantially in parallel with the user's input of additional consonant-vowel pairs. Note ~~Z6~~ FIG. 12, in which the passphrase is confirmed and the private key has been completely unlocked. | Z-1, para. 3, cont'd |
| **0158** | ~~"Secure delay."~~ An element of the inventions is a processing-induced secure delay interposed between an input and output that transforms a given input value into an unpredictable but deterministic output value. [(]Both terms are subject to minor deviations from exactness. An output is "unpredictable" when it is computationally infeasible to predict, given computing resources that could reasonably be expected to be brought to bear against the problem. An output is "deterministic" when it is consistently the result given a particular input, except perhaps for rare errors or imperfections in algorithms that do not have a significant negative impact on performance. | P22/L12-19; P18/L5 *("following terms are said to be elements of the inventions")* |
| **0159** | In system 1500, a passphrase entry user interface 1510 accepts [8] eight [CV] consonant-vowel pairs over 3-10 seconds of user input. | Z-7 *(FIG. 15, ref. 1510 with note directed thereto)* |
| **...** | ~~Set CV:~~ Each pair ~~of consonant, vowel~~ is a set with 13x5=65 members | Z-7 *(Notes directed to connection* |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | ~~of set CV~~, | *FIG. 15, ref. 1510 → ref. 1520)* |
| ... | which block 1520 transforms to a set "A[:]" whose members are the integers from zero to ~~{0,1,...~~ 64[}]. | Z-7 *(FIG. 15, ref. 1320)* |
| ... | The output of block 1520 is a seven-bit word, applied to one input "A" of block 1530. | Z-7 *(connection to FIG. 15, ref. 1330 with width-denoting slash symbol labeled "7")* |
| **0160** | Public key[/] and private key pair generator 1538 connects to block 1532 via user's public key 1540. Block 1532 generates a 121-bit "fingerprint" hash of user's public key 1540 and applies it to a second input "B" of block 1530. | Z-7 *(FIG. 15, ref. 1538 → ref. 1540 → ref. 1532 → ref. 1330, width-denoting slash symbol "121")* |
| **0161** | The (A,B) output of block 1530 is applied to a secure encryption and delay block 1550. | Z-7 *(FIG. 15, ref. 1530 → ref. 1350)* |
| ... | Using an inventive indexed key lookup, block 1550 operates with a number of cycles selected to give [~1/3] about a third-sec[.]ond delay on the user's machine. ~~1/3 x 8 sec ~~~ A third-second delay multiplied by eight (the number of ~~iterate 8~~ times block 1550 iterates, the number of consonant-vowel pairs accepted by user interface 1510) totals about 2.5 seconds and is not very noticeable. | Z-7 *(Notes directed to FIG. 15, ref. 1550)* |
| **0162** | FIG. 16 illustrates a block cipher 1622 that accepts initialization vector input 1612 and (A,B) input 1614. (Initialization vector input 1612 is fixed for the first [CV] consonant-vowel pair[,] processed, and is the result of the previous [CV] consonant-vowel pair for the others.) Block cipher 1622 produces delayed, encrypted output 1642 that is also sent to an XOR 1624 along with output of a key lookup table 1638. | Z-7 *(FIG. 16, refs. 1612, 1614 → ref. 1622; note below ref. 1612; refs. 1622, 1638 → refs. 1642, 1624)* |
| **0163** | Key lookup table 1638 has $2^M$ keys. User system info 1632 is applied to N iterations 1634, the output of which is applied to a counter mod $2^M$ 1636. The M-bit output of counter 1636 is applied to key lookup table 1638. | Z-7 *(FIG. 16, ref. 1632 → 1634 → 1636 → 1638; last connection has width-denoting slash symbol labeled "M")* |
| ... | Regarding key lookup table 1638, truly random is best for preventing an attacker from computing keys on the fly[.] (to avoid using memory). | Z-7 *(Note directed to FIG. 16, ref. 1638)* |
| **0164** | Returning to FIG. 15, public key[/] and private key pair generator | Z-7 *(FIG. 15, ref.* |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
|  | 1538 sends "once" output 1505 to user's private key 1556. (Generator 1538 also receives "many" input 1506 back from user's private key 1556.) | *1538 → ref. 1556, labeled "once"; ref. 1556 → ref. 1538, labeled "many")* |
| ... | User's private key 1556 is very sensitive[,] and never stored anywhere. Symmetric encryption 1552 accepts an input ("once") from user's private key 1556, and another "once" connection goes to encrypted "locked" private key 1554. | *Z-7 (FIG. 15, note to ref. 1556, ref. 1552 ↔ ref. 1556; ref. 1554 → ref. 1552; connection labels "once" & "many")* |
| **0165** | Encrypted "locked" private key 1554 produces an output that symmetric encryp[.]tion 1552 accepts as an input. Symmetric encrypt[.]ion 1552 also accepts as an input "k" the output of secure encryption and delay block 1550, which also also serves as an "I.V." input back to block 1550. | *Z-7 (FIG. 15, ref. 1550 → refs. 1550, 1552; input labels)* |
| ... | Symmetric encrypt[.]ion 1552 produces an output ("many") that goes back to "never stored anywhere" private key 1556. | *Z-7 (FIG. 15, ref. 1552 → ref. 1556; note directed to ref. 1556)* |
| **0166** | User's public key 1540 is publicly available. Encrypted "locked" private key 1554 could be public, too, in view of the passphrase and secure delay security of system 1500. | *Z-7 (FIG. 15, notes directed to refs. 1540, 1554)* |
| **0167** | A secure delay according to various aspects of the present inventions can be applied in other areas than just passphrase security. For example, a hash value can be run through a secure delay to produce a smaller hash value that would be computationally infeasible to derive based on a birthday attack. In one ~~conceived~~ embodiment, a 160-bit hash value is repeatedly run through a secure delay for a predetermined number of iterations that, given a security selection, corresponds to an acceptable unit delay. (An example of an acceptable unit delay is [1] one second.) At the end of each unit delay, a sub-hash is computed from the current output of the secure delay and displayed. | AA-1, para.1 |
| ... | A person wishing to compare hash values can begin comparing a first group of digits corresponding to the first sub-hash after the first unit delay, and while the second unit delay is underway. When the person looks for the second group of digits to compare, the second unit delay (when optimally chosen) is already completed and the third unit delay is underway. | AA-1, para. 2 |
| **0168** | Thus, a securely delayed hash system according to various aspects of the inventions can provide a smaller hash value with the same | AA-1, para. 3 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | security as the larger hash value from which it is derived. The loss in entropy in the smaller value is offset by the computational difficulty (from the secure delay) of obtaining the smaller value. An attacker wishing to find a larger hash value that produces the smaller hash value will need to run the secure delay, on average, N2/2 times with the secure delay computation for each iteration. | |
| **0169** | If T = delay time (on an equivalent processor as that of the legitimate user), then T2 = T*N2/2. If T = 1 second, and the required T2 = 1,000,000 CPU years, then the required N2 [~=] is approximately equal to $2^{21}$, a much smaller value than, say, $2^{160}$. | AA-1, para. 4 |
| **0170** | Since a hash value is not particularly sensitive, it can be sent freely over in secure networks. It is conceivable that an Internet site can be established for quickly computing smaller "sub hashes" based on transmitted hash values through an open-source, standardized secure delay algorithm. However, applicant believes it [is] more likely that the market will demand simplicity and standardization, and an average delay within the reach of the average desktop PC. [(]The remote-computation model may be useful for portable computers, though.[)] | AA-1, last para. |
| **0171** | ~~(As but one example, the applicant views~~ The implementation of a secure delay with a pseudogroup operation~~ as~~ is a particularly advantageous combination according to one aspect of ~~his~~ the inventions. ~~See Appendices K and L for specific disclosure of this particular combination.)~~ | P2/L15-18 |
| ... | As discussed in greater detail below with reference to FIG. 18, a pseudogroup operation according to various aspects of the present inventions dispenses with the need for the modulus p to be fixed with respect to the order of the input and output set | P14/L10-12 |
| ... | in an operation of the type xy modulo p. | P13/L25 |
| ... | Advantageously, the modulus can be chosen as whatever prime number is closest to the set order - above it or below it. As may be better understood with reference to Appendix J-5 of the application as originally filed, for example, two advantageous modifications of the IDEA cipher according to the inventions employ, in one case, a modul[i]us $2^{15}$ greater than $2^{32}$, and in another case, 5 less than $2^{32}$~~,~~ ~~respectively,~~ to encrypt a block of binary data in the set {1, 2, ... $2^{32}$}. TABLE III below lists exemplary prime numbers for various bit lengths. | P14/L12-17; J-2 |
| **0172** | TABLE III *(Table contents omitted.)* | J-2 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| **0173** | FIG. 17 illustrates one exemplary embodiment of a 32-bit secure delay 1700 employing a [α is] generator α of $Z_p$* ~~(fixed)~~, where p is prime and between the values $2^{32}$ - $2^k$ [< p <] and $2^{32}$. | *AC-2 (Note in lower right of page; FIG. 17, ref. 1720 and note above it)* |
| ... | It is assume[s]d that there is a provable prime less than [<] $2^{32}$and within $2^k$ of $2^{32}$, where k is small, [(]4-5[)]. | AC-2 *(Note in lower left of page)* |
| ... | Block 1720 implements the operation $\alpha^x$ mod p, producing a 32-bit output 1703 from a 32-bit input 1701 received via an XOR 1714. | AC-2 *(FIG. 17, ref. 1701 → ref. 714 → ref. 714 )* |
| ... | (The overall output of delay 1700 also feeds back to block 1720 via another 32-bit input 1702 to XOR 1714.) | AC-2 *(FIG. 17, ref. 1702, "OUT")* |
| **0174** | Output 1703 of block 1720 is applied as an input (a,b) to block 1722, which produces separate 16-bit outputs a 1704 and b 1705. (Outputs 1704, 1705 are applied to LUT A 1724 and LUT B 1726, respectively. LUT A,B 1724, 1726 are 64K x 16 and can be the same random mappings of $\{0,1\}^{16}$ or different.) | *AC-2 (FIG. 17, ref. 1722 → refs. 1724, 1726, width-denoting label"16"; note below ref. 1740)* |
| ... | Outputs 1706, 1707 of LUT A 1724 and LUT B 1726 are applied as inputs a, b, respectively, of (a,b) block 1728, which connects to output 1702. | AC-2 *(FIG. 17, refs. 1724, 1726 → refs. 1706, 1707 → ref. 1728 → ref. 1702)* |
| **0175** | Blocks 1732 and 1740 provide an alternative path from output 1710 of XOR 1714 to the overall output 1702 of secure delay 1700. Control block 1715 accepts 1710 as an input and ~~one~~ select[ed]s one of block 1720 and block 1732. When block 1732 is selected, it passes k LSBs of 1710 to LUT C 1740, | AC-2 *(FIG. 17, ref. 1710 → 1720 → 1722 → 1724, 1726 → 1728 is one path; ref. 1710 → 1732 → 1740 is other; "ONE SELECTED" by arrows from ref. 1715)* |
| ... | which provides 32 bits of output at 1702. | AC-2 *(FIG. 17, width-denoting label "32" on ref. 1740 output)* |
| **0176** | LUT C 1740 is $2^k$ x 32. It fills in gaps in the set $\{0,1\}^{32}$ that are not reached by LUT[S (]A[,B)] 1724 and LUT B because (1) p < $2^{32}$[,] and (2) $\alpha^x$ mod p is not selected for x ≥ $2^{32}$ - $2^k$. | AC-2 *(Note at left of drawing)* |
| **0177** | The pseudogroup operation will now be discussedin more detail with reference to FIGS. 18-20. | P13/L24 |
| **0178** | The multiplicative group xy modulo p (where p is prime) has many applications, particularly in the field of cryptography. Any unique | P13/L25-26; P14/L6 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
|  | combination of two multiplicands (i.e., inputs) from a set S:{1,2,...M} multiplied modulo p, where the prime p = M+1, produces a unique result (i.e., output) in the set S. When p is large, the particular combination of inputs used cannot be easily determined based on a given output. The IDEA cipher exploits this property by using a multiplicative group modulo p = $2^{16}$ + 1 (which is prime) along with two other group operations to encrypt binary data in the set {1,2,...$2^{16}$}. |  |
| **0179** | Very few known moduli have the desirable property of being exactly one greater than a power of two. Appendix J-16 of the application as originally filed illustrates the undesirable lack of scalability in the IDEA cipher that results ~~from this fact~~. | P14/L7-9 |
| **0180** | The pseudogroup operation, like the conventional multiplicative group of modulo p, relies on a modulo product of two numbers. This product can be implemented with a regular product followed by a modular reduction (i.e., mod p) operation, or by modular multiplication. Any suitable modular reduction or multiplication technique can be employed. Because p = M +/- k, and k << M, Algorithm 14.47 (see Appendix AL-1,2) from the Handbook of Applied Cryptography (incorporated herein by reference in its entirety) can be advantagously employed. | P16/L8-14 |
| **0181** | FIG. 18 illustrates a pseudogroup operation 1800 from inputs x & y to an output z with $p > 2^N$. | J-1 *(Note at upper right of page, "MODIFIED ⊙"; label "With $p > 2^N$"; input, output labels)* |
| ... | In one type of pseudogroup operation, z = x*y mod p, where p = $2^N$ + k and is [(]prime, preferably the lowest prime greater than $2^N$[)]. | P5/L4-6 |
| ... | Inputs x and y are applied to a "⊙" group operation 1810 and the result is applied to decision block 1822, which determines whether the result is [>] greater than $2^N$[?]. | J-1 *(FIG. 18, ref. 1810→ ref. 1822; labeled as ">$2^N$?" and with outputs labels "NO", "YES")* |
| ... | If it is, the outputof group operation 1810 is re-mapped to a hole | J-1 *(FIG. 18, ref. 1810→1822→1830)* |
| ... | in the output set. A hole is a value that will not ever occur for a particular key, with any input from the set {1,2,...$2^N$}. ~~This option is particularly~~ | P5/L9-10 |
| **0182** | Advantageously, ~~in that it~~ this mapping is fully reversible - each possible input maps to a unique output, and vice versa. | P5/L10-12 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| ... | FIG. 19 illustrates an inverse pseudogroup operation where inputs z & $y^{-1}$ are mapped to an output. Block 1924 calculates holes for y based on input $y^{-1}$. If [is] z is one of the holes[?] calculated, as decided at block 1922, block 1930 does an inverse mapping and the result is applied to a "⊙" operation at 1940, against $y^{-1}$. Otherwise, the "⊙" operation at 1940 uses the non-mapped input z and $y^{-1}$ as its inputs. | J-1 *(Label "INVERSE:"; output labels; input z → ref.s 1924 & 1922; ref. 1924 → ref. 1922 → ref. 1930 → ref. 1940 with input $y^{-1}$; inputs z, $y^{-1}$ → ref. 1940)* |
| **0183** | As discussed above, a pseudogroup with p > M preferably does exception handling by mapping overflowing outputs to holes. Appendix P illustrates results of a holeplotting ~~Octave~~ script written for "Octave," a [(]GNU MATLAB alternative[)]. | P14/L19-21;<br>Q-1, line 3 |
| ... | Pages 1-3 of Appendix P[-1,2,3], illustrate[s] output values (along the row axis perpendicular to the ones and zeros parallel to the handwritten writing) that are holes. There, the holes are values that do not occur within the set S:{[1,2,...64[)]} as the result of a modulo product with modulus > 64, given a particular key value (columns labeled 1-64) within set S. On page 1 of [In] Appendix P[-1], the modulus is 71. On page 2 of [In] Appendix P[-2], the modulus is 67. On page 3 of [In] Appendix P[-3], the modulus is 73. | P14/L21-26 |
| **0184** | The lines on the plots of each page illustrate the deterministic placement of holes given different key values within set S. In the example of ~~Appendix P-1~~ page 1, a key value of 9 will have holes (i.e., non-occurring output values) of 62 (k=1), 53 (k=2), 44 (k=3), 35 (k=4), 26 (k=5), and 17 (k=6). Modulo 71 products that exceed 64 (one for each hole) can be advantageously mapped to a particular hole, depending on the amount of overflow in excess of 64. For example, an output of 65 (k=1) with a key value of 9 can be mapped to the value 62. An output of 66 (k=2) can be mapped to the value 53. Thus, a pseudogroup (p=71) output of 62, given a known key value of 9, can be deterministically converted back to an input value of 23, whose modulo product (unmapped) is 65. | P14/L27-<br>P15/L7 |
| **0185** | A pseudogroup with modulus p < M preferably performs exception handling by permitting input values greater than p-1 to simply pass through unaltered. As illustrated [in] on page one of Appendix I[-1], this exception handling can be made very rare with suitable selection of M and p. Consider the example on[f] page two of Appendix 1[-2], where M = $2^{50}$ and p = $2^{50}$ - 7. (This may or may not be a suitable prime modulus, but it serves the purpose of illustration in this example). Here, the likelihood of processing an input greater than p, given uniform random distribution of input values in (1,2,...$2^{50}$}, | P15/L8-17 |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | appears to be only 0.004 after nearly $10^{12}$ iterations[!]. With this low level of error, the straightforward alternative of simply allowing the erroneous output to occur and permitting conventional error-correction coding to compensate for it becomes attractive. | |
| **0186** | FIG. 20 illustrates a pseudogroup operation from inputs x & y to an output z with $p < 2^N$. | J-1 *(Label "With $p < 2^N$"; input, output labels)* |
| ... | Block 2020 decides if input x is > $2^N - k - 1$[?] and, if so, bypasses "⊙" operation 2010, in which case output z is equal to input x. Otherwise, output z is the result of operation 2010 based on inputs x and y. | J-1 *(FIG. 20, ref. 2020 & ref. 2010 → ouput z; output of ref. 2020 labeled as "YES")* |
| **0187** | ~~Appendix O:~~ TABLE IV below shows a mathematical derivation of exemplary equations for pseudogroup encryption and decryption. | P7/L10-11 |
| **0188** | TABLE IV *(Table contents omitted.)* | O-1 |
| **0189** | When performing a pseudogroup operation with p < M, key values should be less than p. TABLE V below shows an example[s] of such an operation[s], with an unrealistically small but illustrative value[s] of p, ~~are~~ is shown in ~~Appendix J-6~~. | P15/L18-20 |
| **0190** | TABLE V *(Table contents omitted.)* | J-6, bottom table |
| ... | TABLE [I-1] VI below shows results of a ["]pseudogroup["] operation[: p=] with the prime number 11 ~~(prime), m=3~~ and a three-bit set of integers 1-8. | N-1, text at top of page |
| **0191** | TABLE VI *(Table contents omitted.)* | N-1, TABLE I-1 |
| **0192** | TABLE [I-2:] VII below shows key values and holes associated with them for different output values. Each row contains input values producing a given output, except for holes,which are marked with the letter "X." ~~(black squares)~~ Each column contains key values producing outputs for a given input, except for holes. | N-1, text at middle of page |
| **0193** | As mentioned above, a hole is an output value that will not occur for any in the set {1,2, ... $2^m$} of possible key values, given a particular input valued in that set. Note from TABLES VI and VII ~~I-1 and I-2~~ that the output value of 7 does not occur with the key value of 2. | N-1, text at bottom of page |
| **0194** | TABLE VII *(Table contents omitted.)* | N-1, TABLE I-2 |
| **0195** | A cryptographic document destruction | AI-1, title |
| **0196** | [A] system according to various aspects of the invention includes[:] | AI-1, para. 3 |

| Para. | Substitute Specification Text with Changes from Source Material | Source Material |
|---|---|---|
| | an encryption subsystem[;] and a decryption subsystem[,]. The decryption subsystem ~~using~~ uses a temporary key that can be disposed of to make encrypted communications unreadable. | |
| 0197 | An encryption key allows an authorized person to decrypt encrypted communications. For example, an encryption key can be a passphrase, or use a passphrase, known only to a person authorized to decrypt communications. According to various aspects of the present inventions, the decryption key can be destroyed. A passphrase for such a decryption key is preferably forgettable, for example a random alphanumeric string of sufficient length to be secure. Advantageously, the alphanumeric string can be used as a passphrase to open communications or records when is desired to do so and then destroyed and forgotten about when it is no longer desired for such communications or records to ever be decrypted again. | AI-1, para. 4 |
| 0198 | Operation of one embodiment includes (1) writing an electronic mail message to a person; (2) encrypting the communications using an agreed-upon passphrase, preferably an alphanumeric random digit string, for example ~~12~~ twelve digits in length; (3) sending the encrypted message to the recipient; (4) having the recipient type in the passphrase to open the encrypted communications; and then after a predetermined or agreed-upon period of time, (5) having both parties destroy the passphrase (throwing away a POST-IT note upon which the passphrase is written) so that neither the sender nor the recipient can ever decrypt the communication again. Preferably, the passphrase is used only for a short length of time or limited number of times so that it is impossible for either party to remember it. The more random and arbitrary cryptic the passphrase is, the more difficult it will be to ever remember. | AI-1, para. 5;<br>AI-2, partial para. 1 |
| 0199 | Systems according to various aspects of the invention can be useful in the legal profession where sometimes legal professionals are called upon to testify about matters that were assumed to be privileged but the court determines that they are not for whatever reason, as happens in patent practice sometimes. If an attorney or agent has communicated with his client using this system, and the client agrees to destroy the passphrase after the matter is complete, and the device communicated by the attorney or agent is no longer relevant or needed and has been acted upon completely, then it is impossible for any court or any party to discover what to parties discussed. | AI-2, first full para. |
| 0200 | Even if a backup copy of an electronic mail message is found, a court can authorize a cryptananalysis of the message, but if it is encrypted using PGP strong encryption, it would be very difficult, effectively | AI-2, second full para. |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | impossible, for the opposing party to figure out what the message said. | |
| **0201** | Embodiments can be employed in other types of communications that are encoded in digital form so that they can be encrypted. Even handwritten notes can be scanned into digital form and encrypted. | AI-2, third full para. |
| **0202** | Voice messages can be digitized and compressed, entire paper files can be archived by scanning and digitizing, and then encrypting into a single encrypted archive file with a temporary key that can be disposed of after a predetermined time, which can be set by policy, for example one year. | A1-2, fourth full para. |
| **0203** | According to another aspect of the invention, the keys need not be remembered or typed in by a human operator at all. According to this aspect, the key is an actual hardware device that transmits decrypting authorization indicia for a predetermined or agreed-upon period of time and then is incapable of doing so after that. An example of such a key is placed between a conventional PC keyboard and a PC. The device includes circuitry for reading a decryption key code or indicia from another device such as a card having barcodes printed on it or a disposable integrated circuit, which can be made in the form of a key. | A1-2, fifth full para. |
| **0204** | The device can be sold with a number of keys that can be used and disposed of by the user. For example, if the device is sold with twelve keys with refills of ~~12~~ twelve additional keys available by ordering, the user can encrypted archive records every month with a different key and cost a new (different) key away every month. The user may wish to keep the records on file for a period of several months in which case the user will begin using a key one month and then put the key into storage for a couple of months and then toss the key away, destroying it irretrievably after that period of time. | A1-2, last para. |
| **0205** | An embodiment using printed cards is less expensive and the keys can be disposed of more cheaply but it is more prone to unauthorized or inadvertent duplication, in which case the whole purpose of the system might be defeated. The user of such a system needs to take precautions that the keys are never duplicated. | A1-3, first para. |
| **0206** | A database of which files correspond to which temporary keys can be created according to aspects of the invention so that an administrator can look over the list of keys about to expire and ask the persons involved with the effective files whether or not they need information from the files before they are destroyed. Paper documents can be shredded at the same time the keys are destroyed. | A1-3, second para. |

| *Para.* | *Substitute Specification Text with Changes from Source Material* | *Source Material* |
|---|---|---|
| | If the key for a paper file that has a corresponding electronic file is a card, the card can be kept with the paper file and both can be destroyed simultaneously. | |
| **0207** | The key can be distributed from a sender of information to a recipient who is only authorized to access the information for a temporary period of time, for example one or two days. The sender of the information, or provider, can demand the key back after that period of time. In such a system, the key needs to be difficult to duplicate, for example an integrated circuit in the shape of a key. A forgettable password would not work for such a system because the user could write it down without telling the sender, but the forgettable password system works well when both parties, or all parties involved, are in cooperation and consent to destroy the information and the forgettable password. | A1-3, last para. |
| **0208** | In the processf flow illustrated in FIG. 21, a sensitive message 2112, which is never saved, preferably, comes from a sender/originator 2110. Encrypt 2114 uses a temporary key 2122, which is destroyed ~~temp key~~ after a period "X." | AI-4 (*FIG. 21, ref. 2110 → ref. 2112; refs. 2112, 2122 → ref. 2114; misc. notes)* |
| **0209** | Encrypted 2132 goes to backup file, temp. file, output, etc. 2134 and to decrypt 2142, which uses another temporary key 2152, that goes to 2154 after period "X." Sensitive message 2144 from decrypt 2142, which also is preferably never saved, goes to recipient 2160. | AI-4 (*FIG. 21, ref. 2132 → ref. 2134, 2142; ref. 2152 → refs. 2142, 2154; ref. 2142 → ref. 2160; misc. notes)* |
| **0210** | Per another aspect, a hardware electronic record "lock" with a disposable key token is employed, such as a simple paper card with a bar code printed on it. ~~See Appendix AJ.~~ The key token can be thrown away when the "locked" (i.e., encrypted) electronic record is to be purged, ~~by~~ making decryption of it practically impossible. | P13/L15-18 |